

BACHELORTHESIS

Mika Christian Urbach

Planung und Entwicklung eines Plattformunabhängigen Controllers für die netzdienliche Steuerung und die Überwachung von industriellen Photovoltaikanlagen einschließlich der Konzeption und Realisierung eines standardisierten Tools für die projektspezifische Konfiguration.

FAKULTÄT TECHNIK UND INFORMATIK

Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science

Department of Information and Electrical Engineering

Mika Christian Urbach

Planung und Entwicklung eines plattform-unabhängigen Controllers für die netzdienliche Steuerung und die Überwachung von industriellen Photovoltaikanlagen einschließlich der Konzeption und Realisierung eines standardisierten Tools für die projektspezifische Konfiguration.

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr. Holger Gräßner

Zweitprüfer: Nils Heinrich

Abgegeben am 10. Februar 2025

Mika Christian Urbach

Thema der Bachelorthesis: Planung und Entwicklung eines plattformunabhängigen Controllers für die netzdienliche Steuerung und die Überwachung von industriellen Photovoltaikanlagen einschließlich der Konzeption und Realisierung eines standardisierten Tools für die projektspezifische Konfiguration.

Stichworte: Energie-Automation, Photovoltaik, Node-RED, Anlagensteuerung, Raspberry Pi, Open-Source, Energie-Monitoring, Energieerzeugung, Software-Controller

Kurzzusammenfassung: In dieser Arbeit wird ein Controller zur Steuerung und Überwachung von Photovoltaikanlagen praxisnah geplant und entwickelt. Die Entwicklung basiert auf einer Anforderungsanalyse, die vor allem das verpflichtende Netzsicherheitsmanagement umfasst, welche die Erzeugungsleistung situativ anpasst. Zusätzlich werden von dem Controller Betriebsdaten erfasst und einem übergeordneten Energie-Monitoring-System bereitgestellt. Die Entwicklung des Controllers erfolgt plattformunabhängig in Node-RED. Für die Konfiguration des Controllers wird ein Konfigurationstool entwickelt.

Title of the paper: Planning and development of a platform-independent controller for the grid-serving control and monitoring of industrial photovoltaic systems, including the design and implementation of a standardized tool for project-specific configuration.

Keywords: energy automation, photovoltaics, Node-RED, system control, Raspberry Pi, open source, energy monitoring, energy generation, software controller

Abstract: In this thesis, a controller for controlling and monitoring photovoltaic systems is planned and developed. The development is based on a requirements analysis, which primarily includes the mandatory grid security management, which adjusts the generation capacity according to the situation. In addition, the controller records operating data and makes it available to an energy monitoring system. The controller is developed platform-independently in Node-RED. A configuration tool is being developed to configure the controller.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Zielsetzung.....	2
1.3	Vorstellung des Unternehmens.....	4
1.4	Struktur und Arbeitsweise	5
2	Anforderungsanalyse	6
2.1	Beschreibung des begleiteten Projekts	6
2.2	Design-Vorgaben des Unternehmens.....	8
2.3	Regulatorische Anforderungen	9
2.4	Steuerbare Verbrauchseinrichtungen.....	23
3	Grundkonzept.....	24
3.1	Auswahl der Programmierungsumgebung.....	24
3.2	Auswahl der erforderlichen Schnittstellen	26
3.3	Auswahl des Raspberry Pi	27
3.4	Benötigte Daten für die Modbus-Kommunikation	31
3.5	Konfigurierbarkeit von Node-RED	32
3.6	Anforderungen an die Node-RED Programmierung.....	34
4	Programmkonzept.....	36
4.1	Programmstruktur	36
4.2	Unterstützte Geräte	40
4.3	Erstellen des Githubs.....	42
5	Programmierung des PV-Controllers.....	44
5.1	Konfiguration des PV-Controllers	44

5.2	Main Flow.....	50
5.3	Modbus-Kommunikation	54
5.3.1	Modbus-Datenabfrage	54
5.3.2	Modbus Daten senden	60
5.4	Logging der Daten für JEVIs.....	66
5.5	Steuerung der PV-Anlagen	71
5.6	Leistungsreduzierung Verbraucher.....	79
5.7	Klonen des Programms auf Kundengeräte und anderen Controller	81
6	Excel-Konfigurationstool.....	84
6.1	Allgemeiner Aufbau	84
6.2	Deckblatt.....	86
6.3	Projektdaten.....	86
6.4	Allgemeine Angaben.....	88
6.5	Auswahl Wechselrichter	90
6.6	Konfiguration Stromzähler.....	93
6.7	Konfiguration FWA.....	95
6.8	Konfigurationsdatei erzeugen	98
6.9	Konfiguration übertragen	101
7	Erprobung	104
7.1	Testumgebung	104
7.2	Testfälle.....	108
7.3	Durchführung der Erprobung.....	112
7.3.1	Auswertung Testfall 1: Einlesen der Konfiguration.....	112
7.3.2	Auswertung Testfall 2: FWA Nennleistungssteuerung.....	113

7.3.3	Auswertung Testfall 3: FWA Blindleistungssteuerung	115
7.3.4	Auswertung Testfall 4: FWA Messwerte Bezugszähler	117
7.3.5	Auswertung Testfall 5: FWA Messwerte Pyranometer.....	118
7.3.6	Auswertung Testfall 6: FWA Not-Aus.....	119
7.3.7	Auswertung Testfall 7: Begrenzung der Einspeiseleistung.....	120
7.3.8	Auswertung Testfall 8: Reduzierung von Lastspitzen	122
7.3.9	Auswertung Testfall 9: Funktion der JEVIS-Datenbank.....	123
7.4	Ergebniszusammenfassung.....	125
8	Fazit.....	126
9	Ausblick	127
	Literaturverzeichnis	130
	Anhang.....	138
	Anhang A: Blatt E.8.....	138
	Anhang B: Blatt E.9.....	138
	Anhang C: Single-Line-Diagram.....	138
	Anhang D: Quellcode	138
	Anhang E: Konfigurationsdatei	138
	Anhang F: Excel-Konfigurationstool.....	138

Abkürzungsverzeichnis

BHKW	Blockheizkraftwerk
EAAV	Energieanlagen-Anforderungen-Verordnung
E-Fahrzeugen	Elektrofahrzeugen
elek.	Elektrisch
EMS	Energie-Monitoring-System
Envidatec	Envidatec GmbH
FC	Function Code
FRSE	Funkrundsteurempfänger
FWA	Fernwirkanlage
GPRS	General Packet Radio Service
IoT	Internet of Things
IP	Internet Protocol
JEVis	Energie-Monitoring-Software der Firma Envidatec
LS	Leitungsschutzschalter
LaTS	Lasttrennschalter
LINX-120	Loytec LINX-120
MID	Measuring Instruments Directive (deutsch: Messgeräte Richtlinie)
MPP	Maximum Power Point
MS	Mittelspannung
msg	Message
msg.object	message object
msgid	Message ID
NAP	Netzanschlusspunkt
NA-Schutz	Netz- und Anlagenschutz
NELEV	Elektrotechnische-Eigenschaften-Nachweis-Verordnung
NLS	Netzleitstelle
NS	Niederspannung
PV-Anlage	Photovoltaikanlage

RasPi	Raspberry Pi
RevPi	Revolution Pi (Industrie Raspberry Pi der Firma KUNBUS)
RPM	Red Hat Package Manager
RTU	Remote Terminal Unit
SLD	Single-Line-Diagram
SPS	Speicherprogrammierbaren Steuerungen
SSH	Secure Shell
Sun and Charge	Sun and Charge Germany GmbH
TAB	Technische Anschlussbedingungen
TAR	Technische Anschlussregeln
TCP	Transmission Control Protocol
üEKS	übergeordneter Entkopplungsschutz
ÜGS	Übergabestation
USV	Unterbrechungsfreie Stromversorgung
VBA-Makro	Visual Basic for Applications Makro
VDE	Verband der Elektrotechnik Elektronik Informationstechnik e.V.
VDE FNN	Forum Netztechnik/Netzbetrieb im VDE
WEMAG	WEMAG Netz GmbH

Abbildungsverzeichnis

Abbildung 1: Vereinfachte Übersicht zu den wesentlichen Unterschieden des Anschluss- und Nachweisprozesses durch die NELEV-Novelle 2024 und die EAAV zu den bisherigen Prozessen nach den TAR (Quelle: [9], Seite 10)	10
Abbildung 2: Wirkleistung Grenzkurve der Pav,e-Überwachung aus der VDE-AR-N 4105 (Quelle: [11], Abschnitt 5.5.2, Bild 1).....	12
Abbildung 3: Anschlussplan Pav,e-Überwachung, getrennte Abschaltung von Anlagenteilen des EFR4001IP(Quelle: [12], Seite 9).....	13
Abbildung 4: Single-Line-Diagramm des begleiteten Baus.....	16
Abbildung 5: Ausschnitt Blindleistungssteuerung Blatt E.9 der VDE-AR-N 4110 (Quelle: [18], Seite 23).....	20
Abbildung 6: Schematische Darstellung Datenfluss der Fernwirkanlage	22
Abbildung 7: Beispiel msg.object.....	26
Abbildung 8: Konzept zum Einlesen der Konfiguration in Node-RED	33
Abbildung 9: Verwendete Paletten für die Node-RED Programmierung	35
Abbildung 10: Darstellung der Programmstruktur als Flow in Node-RED	38
Abbildung 11: Konfiguration eines GitHub Repository in Node-RED	43
Abbildung 12: Node-RED Config Flow	45
Abbildung 13: Switch Node Split config in flows aus Abbildung 12.....	47
Abbildung 14: Einstellungen eines Change Nodes.....	48
Abbildung 15: Ausschnitt Inverter 1 Flow: Speichern der Konfiguration	49

Abbildung 16: Main Flow	50
Abbildung 17: Switch Node FWA Modbus Enable aus dem FWA Flow	51
Abbildung 18: Function Node Trigger function for Inverter der zyklischen Abfrage der Wechselrichter	52
Abbildung 19: Switch Node für die Abfrage der Wechselrichter	53
Abbildung 20: Modbus-Kommunikation Inverter 1 Flow Teil 1	54
Abbildung 21: Change Node Inv Amps	55
Abbildung 22: Modbus-Kommunikation Inverter 1 Flow Teil 2	57
Abbildung 23: Function Node Inv Amps Scaling	58
Abbildung 24: Modbus-Kommunikation ohne Skalierung im Power Meter 1-3 Flow	59
Abbildung 25: FWA Flow Modbus Daten senden	61
Abbildung 26: Change Node global radiation	63
Abbildung 27: Inverter 1 Flow Modbus Daten senden	65
Abbildung 28: Inverter 1 Flow Logging	66
Abbildung 29: Einstellungen des Logger Node Inverter Yield	67
Abbildung 30: Logging Flow	67
Abbildung 31: Einrichten des RevPi in JEVIs	68
Abbildung 32: Zuordnung einer Datenreihen in JEVIs	70
Abbildung 33: FWA Flow	72
Abbildung 34: PV Control Flow	74

Abbildung 35: Function Node Processing of FWA control signals Teil 1	75
Abbildung 36: Function Node Processing of FWA control signals Teil 2	76
Abbildung 37: Power reduction Flow	79
Abbildung 38: Function Node Consumer Power Reduction.....	80
Abbildung 39: Git Repository klonen auf Kundengeräte	82
Abbildung 40: Konfigurationstool Tabellenblatt: Deckblatt	86
Abbildung 41: Konfigurationstool Tabellenblatt: Projektdaten	87
Abbildung 42: Konfigurationstool Tabellenblatt: Allgemeine Angaben	89
Abbildung 43: Konfigurationstool Tabellenblatt: Auswahl Wechselrichter.....	90
Abbildung 44: Ausschnitt Konfigurationstool Tabellenblatt: WR_Modbusregisterlisten ..	92
Abbildung 45: Konfigurationstool Tabellenblatt: Konfiguration Stromzähler	94
Abbildung 46: Konfigurationstool Tabellenblatt: Konfiguration FWA	96
Abbildung 47: Konfigurationstool Tabellenblatt: Konfigurationsdatei erzeugen.....	98
Abbildung 48: VBA-Makro zum Erzeugen der Konfigurationsdatei.....	99
Abbildung 49: Ausschnitt Konfigurationstool Tabellenblatt: Konfiguration	100
Abbildung 50: Linker Teil des Konfigurationstool Tabellenblatt: Konfiguration übertragen	102
Abbildung 51: Rechter Teil des Konfigurationstool Tabellenblatt: Konfiguration übertragen	103
Abbildung 52: Schematische Darstellung des Testaufbaus	105

Abbildung 53: Test der Modbus Kommunikation des RS-485 zu TCP/IP Umsetzer für das Register mit der Adresse 30775.....	107
Abbildung 54: Nennleistungsvorgabe 60% durch FWA	113
Abbildung 55: Antwort nach Nennleistungsvorgabe 60% durch FWA	114
Abbildung 56:Nennleistungsvorgabe 100% durch FWA	114
Abbildung 57: Antwort nach Nennleistungsvorgabe 100% durch FWA	114
Abbildung 58: Umschaltung des Steuermodus zur Blindleistungsvorgabe durch NLS und Blindleistungsvorgabe von 20 %.....	115
Abbildung 59: Antwort Umschaltung des Steuermodus zur Blindleistungsvorgabe durch NLS und Blindleistungsvorgabe von 20 %	116
Abbildung 60: Umschaltung des Steuermodus zurück in den Kennlinienbetrieb	116
Abbildung 61: Antwort Umschaltung des Steuermodus zurück in den Kennlinienbetrieb	117
Abbildung 62: Test der Function Nodes zur Berechnung der Phasenverschiebung $\cos(\phi)$ und Q/P installiert in %.....	118
Abbildung 63: Globalstrahlung des Pyranometers vorgeben	119
Abbildung 64: Antwort auf Vorgabe der Globalstrahlung des Pyranometers	119
Abbildung 65: FWA Not-Aus-Signal vorgeben.....	119
Abbildung 66: Gespeichertes Not-Aus-Signal auf dem PV-Controller	119
Abbildung 67: Begrenzung der Einspeiseleitung bei 100 % Vorgabe der FWA.....	120
Abbildung 68: Antwort Begrenzung der Einspeiseleitung	121

Abbildung 69: Antwort Begrenzung der Einspeiseleitung bei unverändert zu hoher Einspeiseleistung	121
Abbildung 70: Antwort Begrenzung der Einspeiseleitung bei Einspeiseleistung unterhalb der Grenze zur Leistungsreduzierung.....	121
Abbildung 71: Begrenzung der Einspeiseleitung bei Wirkleistungsvorgabe durch FWA	122
Abbildung 72: Antwort Begrenzung der Einspeiseleitung bei Wirkleistungsvorgabe durch FWA	122
Abbildung 73: Intervall und Logger Nodes des Inverter 1 Flows	124

Tabellenverzeichnis

Tabelle 1: Übersicht der Leistungen der vor Ort befindlichen Erzeugungsanlagen	7
Tabelle 2: Datenpunktliste der Fernwirkanlage (Quelle: [21], Seite 10)	19
Tabelle 3: RevPi Modelle und Preise [40]	28
Tabelle 4: Übersicht RevPi Erweiterungsmodule [40]	29
Tabelle 5: Zwei Beispiel Zeilen der Konfigurationsdatei	46
Tabelle 6: Ergebnis Testfall 9: Reduzierung der Lastspitzen	123

Formelverzeichnis

Formel1: Berechnung der kumulierten Leistung aller Erzeugungseinheiten und Speicher am NAP bzw. die installierte Wirkleistung aller Erzeugungsanlagen und Speicher	10
Formel 2: Leistungsverhältnis Anschlussleistung zu installierter Wirkleistung aller Erzeugungsanlagen und Speicher	11
Formel 3: Wirkleistungs-Grenzkurve $P_{av,e}$ -Überwachung aus der VDE-AR-N 4105 [11]..	12

1 Einleitung

Aus Gründen der besseren Lesbarkeit wird in dieser Bachelorarbeit das generische Maskulinum verwendet. Sämtliche verwendete Personenbezeichnungen beziehen sich auf alle Geschlechter.

Der Anhang zur Arbeit befindet sich auf DVD und kann beim Erstgutachter eingesehen werden.

1.1 Motivation

Der Klimawandel stellt die globale Gesellschaft vor eine der größten Herausforderungen der Menschheit, denn Extremwetterereignisse nehmen zu und die steigenden Meeresspiegel bedrohen bereits jetzt Menschen in Küstenregionen weltweit. Um den Klimawandel und dessen Folgen abzuschwächen, muss sich der globale CO₂-Ausstoß deutlich reduzieren. Deshalb ist die sogenannte Energiewende von zentraler Bedeutung, damit Industriestaaten wie Deutschland ihre Treibhausgasemission weiter reduzieren und die gesetzten Klimaziele erfüllen [1].

Durch den Ausbau der erneuerbaren Energien steigt die Anzahl von Erzeugungsanlagen im öffentlichen Versorgungsnetz stark an. Dies hat zur Folge, dass die Netzkomplexität der Versorgungsnetze ansteigt und für Erzeugungsanlagen eine Steuerung durch den Versorgungsnetzbetreiber notwendig wird. Diese wird je nach Leistung der Erzeugungsanlage in unterschiedlichem Umfang ausgeführt. Mit dieser ist es dem Versorgungsnetzbetreiber möglich, die aktuelle Einspeiseleistung der Erzeugungsanlage abzufragen und zu reduzieren. Diese Steuereingriffe sind notwendig, um die Stabilität des Versorgungsnetzes zu gewährleisten und kritische Situationen durch gezielte Eingriffe abzuwenden.

Durch eine intelligente Steuerung lassen sich die Anforderungen des Netzbetreibers umsetzen und zeitgleich lässt sich der Eigenverbrauch des selbst produzierten Stroms erhöhen, z.B. durch das sogenannte Überschussladen von Elektrofahrzeugen. Die für die

Steuerung benötigten Daten können zusätzlich in einem Energie-Monitoring-System (EMS) genutzt werden. Mit einem EMS lassen sich die Energieverbräuche aufbereiten und es können Maßnahmen zur Effizienzsteigerung abgeleitet werden. Eine mögliche Maßnahme wäre z.B. das gezielte Abschalten von Anlagenteilen, wenn diese stillstehen. Typischerweise wird nach der Einführung des EMS durch einfache und kostengünstige Maßnahmen der Energiebedarf um bis zu 15 % reduzieren [2].

Die Motivation dieser Arbeit besteht darin, den Ausbau von Photovoltaikanlagen (PV-Anlagen) zu vereinfachen und die netzkonforme Steuerung von PV-Anlagen universell umzusetzen. Zusätzlich wird eine preiswerte Möglichkeit geschaffen, die an den Controller angeschlossenen Geräte in ein EMS einzubinden. Diese Steuerung soll einen positiven Beitrag zur Energiewende leisten, um den Klimawandel weiter abzubremesen.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die prototypische Entwicklung eines Controllers, welcher die netzdienliche Steuerung und die Überwachung von industriellen PV-Anlagen übernimmt. Die Anforderungen an den Controller werden dabei Anhand von realen Projektbedingungen einer im Bau befindlichen 250 kWp PV-Anlage abgeleitet. Für die Erprobung des Controllers wird eine Testumgebung, bestehend aus einem Aufbau aus Hard- und Software sowie definierten Testfällen, erarbeitet. Dies hat den Vorteil, dass die Fertigstellung dieser Arbeit nicht abhängig vom Baufortschritt des begleiteten Projekts ist und auch Störfälle getestet werden können, die im Regelbetrieb nicht auftreten sollten.

Für das EMS sollen durch den Controller die Energie- und Zustandsdaten der verbundenen Geräte erfasst und in einem Ringspeicher gespeichert werden. Die Daten aus dem Ringspeicher werden der Energiemonitoring Software JEVIs, der Firma Envidatec, zugänglich gemacht, sodass diese für das Energiemanagement verwendet werden können.

Neben der Steuerung von PV-Anlagen soll für steuerbare Verbrauchseinrichtungen wie z.B. Ladeinfrastruktur, eine Empfehlung zur Leistungsreduzierung aus Sicht der Anforderungen des Netzes auf dem Controller ausgegeben werden, zur Vermeidung von Spitzenlasten.

Als zentrale Zielsetzung wurde durch die Envidatec GmbH (Envidatec) vorgegeben, dass bei der Entwicklung des Controllers insbesondere darauf geachtet wird, dass der Controller plattformunabhängig ist und auf eine Vielzahl anderer Plattformen migriert werden kann [3]. Dadurch soll unter anderem sichergestellt werden, dass der Controller zukunftsfähig bleibt und für mindestens zehn Jahre weiter genutzt werden kann. Um zukunftsfähig zu bleiben, muss der Aufbau der Programmierung des Controllers so designt werden, dass er in Zukunft um weitere Funktionen erweitert werden kann. Für die Durchführung dieser Arbeit wurden noch weitere Anforderungen durch die Envidatec vorgegeben, diese sind im Abschnitt 2.2 Design-Vorgaben des Unternehmens beschrieben. Dabei wurden folgende Schwerpunkte festgelegt.

Entwicklung des Controllerdesign

- Analyse der Anforderungen anhand eines begleiteten Projekts, dem Bau einer 250 kW PV-Anlage
- Entwicklung eines plattformunabhängigen Gesamtkonzepts.
- Auswahl einer geeigneten Industrie-Hardware sowie der Programmierungsumgebung zur Implementierung des Controllers
- Planung und Spezifikation der Programmabläufe
- Aufbau einer Testumgebung und Definition von Testfällen zur Erprobung
- Berücksichtigung der zukünftigen Pflege und Erweiterbarkeit des Controllers

Implementierung

- Prototypische Umsetzung der Anforderungen auf einer industriellen Hardware
- Integration einer konfigurierbaren Kommunikationsschnittstelle
- Speicherung der Energie- und Betriebsdaten im Ringspeicher zur Weitergabe an das JEVIs Energie-Monitoring-System
- Entwicklung eines standardisierten Excel-Dokuments zur Konfiguration anhand spezifischer Projektanforderungen
- Erprobung des Controllers in der Testumgebung anhand der spezifizierten Testfälle

Der Controller wird unter dem Arbeitstitel „PV-Controller“ entwickelt. Dieser Arbeitstitel wird im weiteren Verlauf der Arbeit verwendet, um sich auf den zu entwickelnden Controller zu beziehen.

1.3 Vorstellung des Unternehmens

Diese Bachelorarbeit ist in Zusammenarbeit mit der Envidatec GmbH entstanden. Die Envidatec wurde 2001 in Hamburg gegründet. Die Firma kann auf mehr als 20 Jahre Erfahrung mit mehr als 750 Projekten in über 40 Ländern zurückblicken. Die Envidatec bietet eine Vielzahl an Dienstleistungen im Bereich des Energie-Managements an. Eine der Kernkompetenzen der Firma ist die Durchführung von Energieaudits nach DIN 16247 sowie die Einführung von Energiemanagementsystemen nach ISO 50001 [4].

Envidatec entwickelt seit dem Jahr 2003 die Energie-Monitoring-Software JEVIs. Mit der Software lassen sich energierelevante Daten erfassen, überwachen und optimieren. Die Energiedaten werden fortlaufend aufgezeichnet und für die Nutzer aufbereitet. Die erfassten Daten lassen sich auf Dashboards und in Analysen vielfältig auswerten. Zudem können Alarmer konfiguriert werden, die z.B. bei einer Störung die zuständigen Stellen informieren. Die Software findet vorwiegend Anwendung bei Kunden im produzierenden Gewerbe sowie bei Betreibern großer Liegenschaften. Im Jahr 2011 wurde JEVIs unter einer

Open-Source-Lizenz veröffentlichen. Damit ist JEVIS die erste Open-Source-Software für industrielles Energiemonitoring [5].

Weitere Kompetenzen der Envidatec GmbH sind:

- Erneuerbare Energien
- Nachhaltigkeitskonzepte
- Energieeffizienz und Energiemanagement
- Ladeinfrastruktur für die Elektromobilität

Im Jahr 2023 wurde die Sun and Charge Germany GmbH (Sun and Charge) als Spin-off der Envidatec gegründet. Grund hierfür war die steigende Nachfrage von Envidatec Kunden nach PV-Anlagen und Ladeinfrastruktur für die Elektromobilität. Die Sun and Charge bietet Kunden ein Komplettpaket an, von der Planung über die Finanzierung, Montage und Inbetriebnahme bis hin zur technischen und kaufmännischen Betriebsführung. Ihre Kernkompetenz ist der Ausbau von PV-Anlagen und Ladeinfrastruktur insbesondere mit einem Fokus auf überdachte Solarparkplätze [6].

1.4 Struktur und Arbeitsweise

Die Struktur der Arbeit teilt sich in mehrere Teile auf. In der Anforderungsanalyse wird zunächst der begleitete Bau und die Design-Vorgaben der Envidatec beschrieben. Anhand des begleiteten Projekts werden die regulatorischen Anforderungen an die PV-Anlage sowie an den PV-Controller betrachtet. Anschließend wird ein Grundkonzept für den PV-Controller erarbeitet. Dabei wird die Programmierungsumgebung sowie ein geeigneter Controller für die Implementierung ausgewählt. Ebenfalls wird herausgearbeitet, welche Daten für eine Modbus-Kommunikation benötigt werden und wie sich die Programmierungsumgebung konfigurieren lässt. Vor dem Beginn der Programmierung wird im Abschnitt Programmkonzept der Programmablauf mit einem besonderen Fokus auf der Erweiterbarkeit geplant.

Im Anschluss wird die Programmierung des PV-Controllers durchgeführt und seine Arbeitsweise beschrieben. Anschließend wird das Konfigurationstool beschrieben, mit dem eine geführte Konfiguration des PV-Controllers vorgenommen werden kann. Dann wird der PV-Controller anhand von Testfällen in einer selbsterrichteten Testumgebung auf seine Funktion überprüft und es wird ein Fazit gezogen sowie ein kleiner Ausblick auf mögliche Weiterentwicklungen des PV-Controllers gegeben.

2 Anforderungsanalyse

Im Rahmen dieser Bachelorarbeit wurde die Planung und der Bau einer PV-Anlage mit ca. 250 kWp begleitet. Dabei wurden anhand des Projekts die Anforderungen an den PV-Controller definiert. Zur Bestimmung weiterer Anforderungen wurden Gespräche mit der technischen Leitung der Envidatec sowie der Geschäftsführung der Sun and Charge geführt. In den nachfolgenden Abschnitten werden die Ergebnisse der durchgeführten Anforderungsanalyse dargestellt. Dabei werden technische Aspekte sowie auch unternehmensspezifische Vorgaben betrachtet. Die Anforderungsanalyse stellt die zentrale Grundlage für die Entwicklung des PV-Controllers in dieser Arbeit dar.

2.1 Beschreibung des begleiteten Projekts

Anhand des begleiteten Projekts lassen sich die Anforderungen an den PV-Controller ableiten. An dem Industriestandort befindet sich eine Lebensmittelproduktion. Die hergestellten Lebensmittel müssen durchgängig gekühlt werden. Dies hat zur Folge, dass der Strombedarf, auf Grund der Kälteanlagen auf einem konstant hohen Niveau ist. Der Jahresverbrauch an Strom beläuft sich auf ca., 4 GWh. Der produzierte Strom soll verwendet werden, um durch Eigennutzung den Netzbezug zu reduzieren. Aus diesem Grund wurde sich als Betreibermodell für die Überschusseinspeisung entschieden, bei dem der selbstproduzierte Strom selbst verbraucht wird und nur der überschüssige Strom in das Versorgungsnetz eingespeist wird. Im Gegensatz dazu würde in einer Volleinspeiseanlage der

gesamte produzierte Strom in das Versorgungsnetz einspeist werden. Im Regelbetrieb wird der Fall, dass überschüssiger Strom in das Netz eingespeist wird, nur vorkommen, wenn Teile der Produktion stillstehen. Durch den Neubau einer weiteren Produktionshalle, auf der sich ebenfalls die neue PV-Anlage befindet, wird der Energiebedarf des Industriestandortes weiter steigen. Der Industriestandort ist über einen Netzanschlusspunkt (NAP) mit einer Anschlussleistung von 1,2 MW an die Mittelspannung angeschlossen.

Es befinden sich, neben der im Bau befindlichen PV-Anlage, drei weitere Erzeugungsanlagen hinter demselben Netzanschlusspunkt. Eine Übersicht der Erzeugungsanlagen ist in Tabelle 1 abgebildet. Die Wärmeproduktion des Blockheizkraftwerks (BHKW) wird nicht betrachtet, da für etwaige Anforderungen an den Netzanschluss nur die elektrische Leistung von Interesse ist. Zudem wird das BHKW bereits ab einem Netzbezug von 25 kW von einer bestehenden BHKW-Steuerung gedrosselt und ggf. abgeschaltet, sofern der Netzbezug weiter unterschritten wird.

Tabelle 1: Übersicht der Leistungen der vor Ort befindlichen Erzeugungsanlagen

Erzeugungsanlage	Wechselrichter	AC-Leistung	DC-Leistung
PV-Anlage (Neubau)	SMA SUNNY TRIPOWER CORE 2 (STP 110-60)	2x 110 kW	250 kWp
PV-Anlage (Bestand)	SMA SUNNY TRIPOWER CORE 2 (STP 110-60)	110 kW	135 kWp
PV-Anlage (Bestand)	SMA SUNNY TRIPOWER CORE 1 (STP 50-41)	50 kW	60 kWp
BHKW (Bestand)	/	50 kW	/

Die bestehenden PV-Anlagen verfügen bereits über ein Netzsicherheitsmanagement, welches für die neu zu errichtende PV-Anlage angepasst werden muss. Welche Funktion das Netzsicherheitsmanagement übernimmt und welche Anpassungen vorgenommen werden müssen, wird in Abschnitt 2.3 Regulatorische Anforderungen herausgearbeitet.

2.2 Design-Vorgaben des Unternehmens

Für die Entwicklung des PV-Controllers wurden durch die Envidatec Designvorgaben gemacht, die im folgenden Abschnitt näher erläutert werden.

Der PV-Controller wird hardwareunabhängig entwickelt, um die Abhängigkeit hinsichtlich der Preisentwicklung und der Verfügbarkeit einzelner Hersteller und Modelle programmierbarer Controller zu reduzieren. Ereignisse der jüngsten Vergangenheit, wie z.B. die Blockade des Suezkanals oder die Chipkrise haben gezeigt, dass dies für eine wirtschaftliche Nachhaltigkeit von Produkten wesentlich ist. Durch die hardwareunabhängige Entwicklung des PV-Controllers kann er bei Bedarf auf eine neue Plattform migriert werden. Dadurch wird sichergestellt, dass der zu entwickelnde PV-Controller zu marktgängigen Preisen angeboten werden kann [3].

Bei der Auswahl der Programmiersprache ist darauf zu achten, dass diese über eine Open-Source-Community verfügt [7]. Dadurch kann auf eine Vielzahl bestehender Funktionen zurückgegriffen werden. Der PV-Controller soll im Rahmen der prototypischen Entwicklung auf einem Raspberry Pi (RasPi) implementiert werden. Dabei ist das Modell nicht vorgegeben, es sollte jedoch auf die Industrietauglichkeit geachtet werden [7]. Um eine einfache Konfiguration des PV-Controllers zu erlauben, soll die Konfiguration des PV-Controllers mit Hilfe eines Excel-Tools erstellt werden, sodass auf dem PV-Controller der Konfigurationsaufwand für weitere Projekte möglichst geringgehalten wird und ohne tiefe Programmierkenntnisse vorgenommen werden kann [3]. Die Datenerfassung für das Energie-Monitoring in JEVIS soll viertelstündlich erfolgen. Diese Auflösung wird typischerweise für das Energie-Monitoring verwendet und ist das Standard-Erfassungsintervall für Lastdaten im Strommarkt [7]. Zur Kommunikation des PV-Controllers mit

angeschlossenen Geräten soll nach Möglichkeit eine Modbus TCP Verbindung genutzt werden. Falls Geräte lediglich über Modbus RTU auf Basis von RS-485 kommunizieren, soll ein RS-485 zu TCP/IP Umsetzer der Firma Waveshare verwendet werden [7] [8]. Dies hat den Vorteil das keine aufwändige RS-485 Busverkabelung hergestellt werden muss, sondern das meist vorhandene TCP/IP-Netzwerk genutzt werden kann.

Das standardisierte Excel-Tool zur projektspezifischen Konfiguration (Excel-Konfigurationstool) des PV-Controllers soll so entwickelt werden, dass es auf jedem gängigen Büro-PC genutzt werden kann [7].

2.3 Regulatorische Anforderungen

Die regulatorischen Anforderungen aus geltenden Normen und Gesetzen werden anhand des begleiteten Projekts, dem Bau einer 250 kWp PV-Anlage, für die Entwicklung des PV-Controllers abgeleitet. Zunächst müssen hierfür die Bestandsituation sowie die mit dem Bau der neuen PV-Anlage erweiterten Anforderungen für die netzseitige Inbetriebnahme der PV-Anlage ermittelt werden. Diese Anforderungen lassen sich an der nachfolgenden Abbildung 1, aus der VDE-Anwendungshilfe „Vereinfachter Anschluss und Nachweis von Erzeugungsanlagen und Speichern mit Netzanschluss in der Mittel- und Hochspannung“ [9], ablesen.

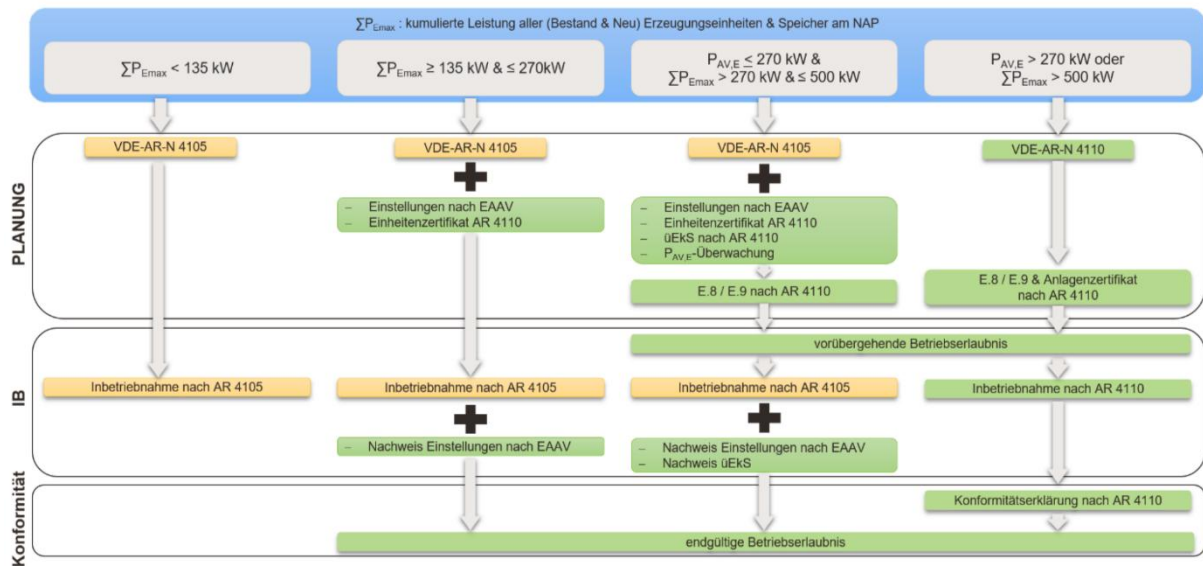


Abbildung 1: Vereinfachte Übersicht zu den wesentlichen Unterschieden des Anschluss- und Nachweisprozesses durch die NELEV-Novelle 2024 und die EAAV zu den bisherigen Prozessen nach den TAR (Quelle: [9], Seite 10)

Um zu bestimmen, welche Anforderungen an die im Bau befindliche PV-Anlage gestellt werden, muss zunächst die kumulierte Leistung aller Erzeugungseinheiten und Speicher am Netzanschlusspunkt $\sum P_{Emax}$ bestimmt werden. Neben der im Bau befindlichen PV-Anlage befinden sich bereits zwei weitere PV-Anlagen mit 110 kW und 50 kW Wechselrichter-Leistung sowie ein BHKW mit 50 $kW_{elek.}$ im Bestand. Damit ergibt sich

$$\begin{aligned} \sum P_{Emax} &= P_{inst} = P_{PV,110kW} + P_{PV,50kW} + P_{BHKW} + P_{PV,220kW} \\ &= 110 \text{ kW} + 50 \text{ kW} + 50 \text{ kW} + 220 \text{ kW} = 430 \text{ kW} \end{aligned}$$

Formel1: Berechnung der kumulierten Leistung aller Erzeugungseinheiten und Speicher am NAP bzw. die installierte Wirkleistung aller Erzeugungsanlagen und Speicher

und folglich muss die dritte Spalte der Abbildung 1 betrachtet werden. Um die PV-Anlage nach den Anforderungen der dritten Spalte in Betrieb setzen zu können, darf die mit dem Versorgungsnetzbetreiber vereinbarte Anschlusswirkleistung (maximale Einspeiseleistung in das Versorgungsnetz) $P_{AV,E}$ maximal 270 kW betragen.

Nach §3 Energieanlagen-Anforderungen-Verordnung (EAAV) ist für den Bau der PV-Anlage trotz des Netzanschlusses in der Mittelspannung die Anschlussregel VDE-AR-N 4105 „Erzeugungsanlagen am Niederspannungsnetz“ anzuwenden [9] [10]. In §3 EAAV werden ergänzend von der VDE-AR-N 4105 folgende Anforderungen gestellt:

- Falls ein Frequenzvermögen vorhanden ist, so ist dieses auszunutzen bis 52,5 Hz nach der P(f)-Kennlinie der VDE-AR-N 4110.
- Ein zweistufiger Überfrequenzschutz ist mit den Einstellwerten der VDE-AR-N 4110 auszuführen.
- Eine Inselnetzerkennung der Erzeugungsanlage ist zu deaktivieren.
- Ein übergeordneter Entkuppelungsschutz (üEKS) nach der VDE-AR-N 4110 ist zu installieren oder eine alternative Einrichtung, welche von der VDE FNN (Forum Netztechnik/Netzbetrieb im VDE) als gleichwertig anerkannt ist.
- Eine $P_{AV,E}$ -Überwachung (Einspeisebegrenzung) ist zu installieren.

Zunächst wird die $P_{AV,E}$ -Überwachung näher betrachtet.

Für die $P_{AV,E}$ -Überwachung ist ein geeignetes System anhand des Leistungsverhältnis von $\frac{P_{AV,E}}{\sum P_{E\max}}$ zu bestimmen. Das Leistungsverhältnis beträgt bei dem begleiteten Bau:

$$\frac{P_{AV,E}}{\sum P_{E\max}} = \frac{P_{AV,E}}{P_{inst}} = \frac{270 \text{ kW}}{430 \text{ kW}} = 0,63 = 63 \%$$

Formel 2: Leistungsverhältnis Anschlussleistung zu installierter Wirkleistung aller Erzeugungsanlagen und Speicher

Bei einem Leistungsverhältnis von $\frac{P_{AV,E}}{\sum P_{E\max}} < 0,54$ ist eine $P_{AV,E}$ -Überwachung nach dem FNN-Hinweis „ $P_{AV,E}$ -Überwachung bei Anschlüssen am Mittel- und Hochspannungsnetz“ zu installieren [9] [10].

Bei dem Leistungsverhältnis des begleiteten Baus von $\frac{P_{AV,E}}{\sum P_{E\max}} \geq 0,54$ ist eine $P_{AV,E}$ -Überwachung nach der VDE-AR-N 4105 Abschnitt 5.5.2 gefordert [9] [10]. Aus der VDE-AR-N 4105 Abschnitt 5.5.2 werden noch weiter Anforderungen an die vereinbarte Anschlusswirkleistung $P_{AV,E}$ gestellt. Diese muss mindestens 60 % der installierten Wirkleistung P_{inst}

betragen [11]. In der Formel 2 wurde dieses Verhältnis bereits berechnet. Die Anschlusswirkleistung $P_{AV,E}$ beträgt 63 % der installierten Wirkleistung P_{inst} , damit ist die Forderung aus der Anwendungsregel erfüllt.

Bei einer Überschreitung der Anschlusswirkleistung $P_{AV,E} = 270 \text{ kW}$ muss die Erzeugungsleistung gedrosselt werden oder ein Verbraucher dazugeschaltet werden, um die Anschlusswirkleistung $P_{AV,E}$ zu reduzieren und die PV-Anlagen weiter netzkonform zu betreiben. Für die $P_{AV,E}$ -Überwachung wird eine Wirkleistungs-Grenzkurve vorgegeben, die beim Überschreiten der Anschlusswirkleistung $P_{AV,E}$ beachtet werden muss. Diese ist in der Abbildung 2 abgebildet und folgt im exponentiellen Verlauf der folgenden Funktion:

$$P_{AV,E}(t) = 0,67 \cdot e^{-1,05 \cdot (t-0,8)} + 1$$

Formel 3: Wirkleistungs-Grenzkurve $P_{AV,E}$ -Überwachung aus der VDE-AR-N 4105 [11]

Beim Überschreiten dieser Wirkleistungs-Grenzkurve sind die Erzeugungsanlagen innerhalb von 200 Millisekunden abzuschalten [11].

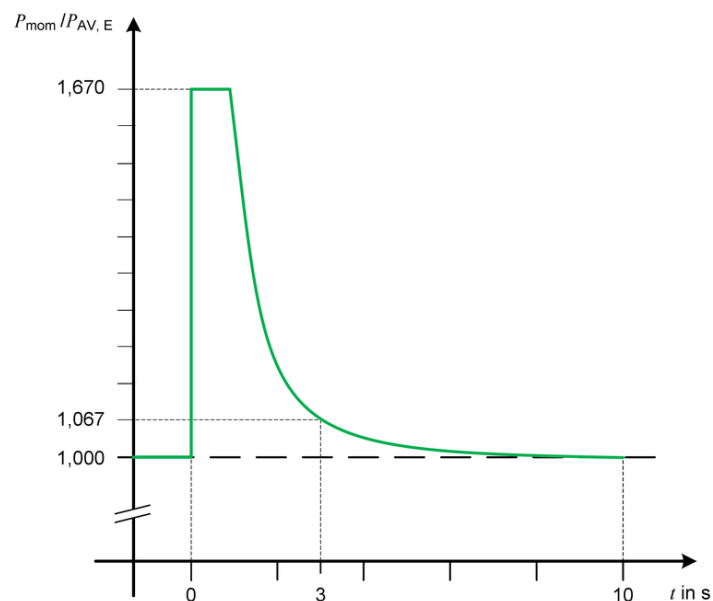


Abbildung 2: Wirkleistung Grenzkurve der $P_{AV,E}$ -Überwachung aus der VDE-AR-N 4105 (Quelle: [11], Abschnitt 5.5.2, Bild 1)

Der im Bau verwendete Regler ERF4001IP der Firma Ziehl zur $P_{AV,E}$ -Überwachung unterstützt keine Leistungsreduzierung an den Erzeugungseinheiten. Der Regler schaltet beim

Überschreiten der Anschlusswirkleistung $P_{AV,E}$ weitere Verbraucher ein, um die Anschlusswirkleistung $P_{AV,E}$ zu reduzieren. Beim Überschreiten der Wirkleistungs-Grenzkurve schaltet der Regler Teile der Erzeugungsanlagen ab, sodass maximal die Anschlusswirkleistung $P_{AV,E}$ in das Netz eingespeist werden kann [12]. Dies lässt sich ebenfalls an dem Anschlussplan des Reglers in Abbildung 3 erkennen.

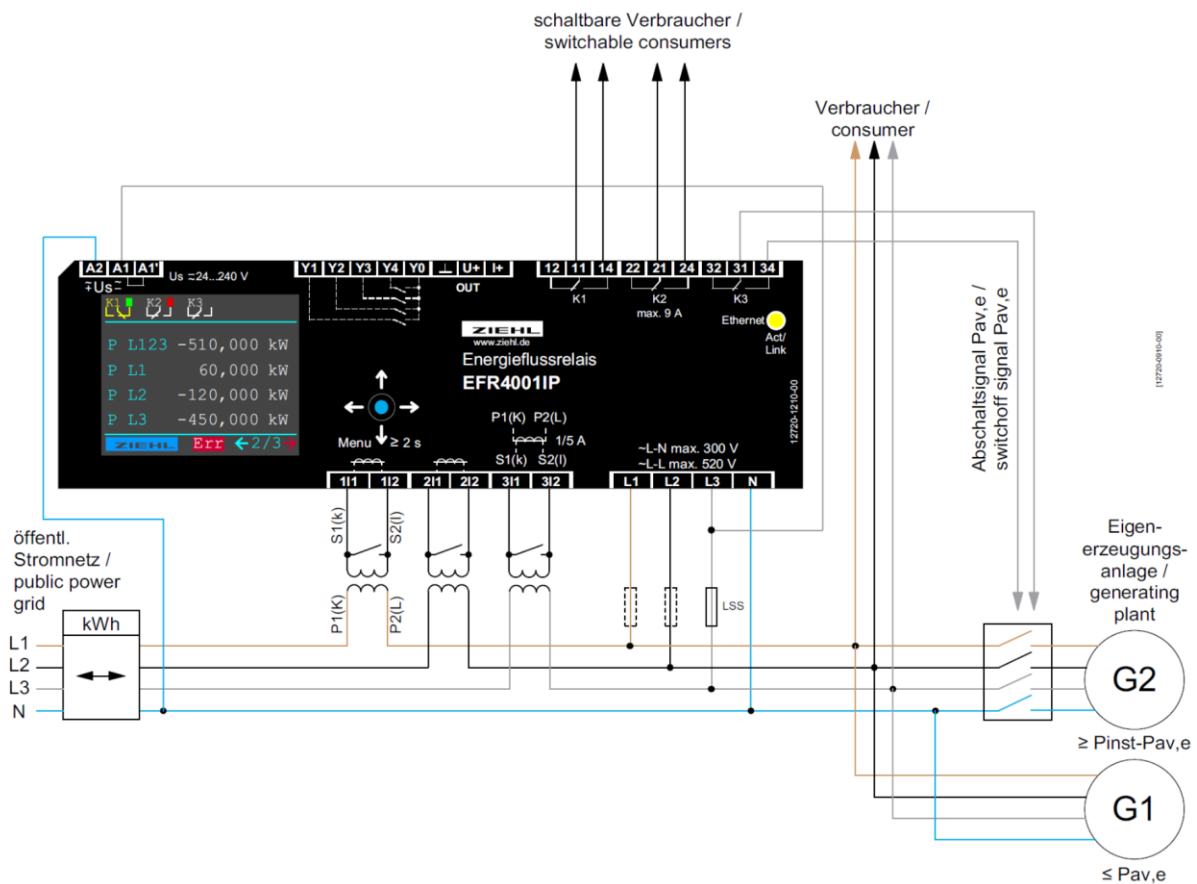


Abbildung 3: Anschlussplan $P_{av,e}$ -Überwachung, getrennte Abschaltung von Anlagenteilen des EFR4001IP (Quelle: [12], Seite 9)

Neben der $P_{AV,E}$ -Überwachung ist ein vorhandenes Frequenzvermögen bis 52,5 Hz nach der P(f)-Kennlinie der VDE-AR-N 4110 auszunutzen. Diese Kennlinie wird bei der Kommissionierung der Wechselrichter über die Einstellwerte in den Wechselrichtern hinterlegt. Die Wechselrichter nutzen so das vorhandene Frequenzvermögen bis 52,5 Hz aus und es müssen keine weiteren Einstellungen vorgenommen werden [13] [14].

Die Forderung aus §3 EAAV nach einem zweistufigen Überfrequenzschutz und einem und einem üEKS wird im Folgenden mitbetrachtet. Auf Grund des Anschlusses an das Versorgungsnetz der WEMAG, müssen zusätzlich die technischen Anschlussbedingungen (TAB) der WEMAG beachtet werden. Die TAB ergänzen jeweils die Anschlussregeln VDE-AR-N 4105 und VDE-AR-N 4110, auf Grund dessen das beide Anschlussregeln Anwendung finden müssen die TAB Niederspannung (NS) und die TAB Mittelspannung (MS) mitbetrachtet werden [15] [16].

Aus der VDE-AR-N 4105 werden weitere Anforderungen an die Schutzeinrichtungen gestellt. So wird in Abschnitt 6.1 der Anwendungsregel die Anforderung gestellt, einen Netz- und Anlagenschutz (NA-Schutz) zu realisieren [11]. Der NA-Schutz trennt bei Netzstörungen, die Erzeugungsanlagen vom Netz, um Schäden an dem Netz und der Erzeugungsanlagen zu verhindern. Auf Grund dessen, dass die Summe der maximalen Scheinleistung der Erzeugungsanlagen am NAP $\sum S_{Amax}$ größer als 30 kVA ist, muss ein zentraler NA-Schutz am zentralen Zählerplatz installiert werden. Nach dem Abschnitt 6.5.1 der VDE-AR-N 4105 muss der NA-Schutz folgende Funktionen beinhalten [11]:

- Spannungsrückgangsschutz (Unterspannungsschutz) $U <$ und $U <<$
- Spannungssteigerungsschutz (Überspannungsschutz) $U >$ und $U >>$
- Frequenzrückgangsschutz (Unterfrequenzschutz) $f <$
- Frequenzsteigerungsschutz (Überfrequenzschutz) $f >$

In §3 EAAV wird zusätzlich ein zweistufigen Überfrequenzschutz und ein üEKS nach der VDE-AR-N 4110 gefordert. Dafür müssen Entkopplungsschutzeinrichtungen am NAP und an den Erzeugungseinheiten installiert sein. Dabei kann der Entkopplungsschutz auch in der Erzeugungseinheit integriert sein. Der üEKS beinhaltet folgende Funktionen [17]:

- Spannungsrückgangsschutz (Unterspannungsschutz) $U <$ und $U <<$
- Spannungssteigerungsschutz (Überspannungsschutz) $U >$ und $U >>$
- Frequenzrückgangsschutz (Unterfrequenzschutz) $f <$
- Frequenzsteigerungsschutz (Überfrequenzschutz) $f >$ und $f >>$
- Q-U-Schutz

Es fällt auf, dass die geforderten Funktionen des NA-Schutzes vollständig vom üEKS abgebildet werden. Aus diesem Grund übernimmt der üEKS den geforderten NA-Schutz. Neben dem üEKS wird noch ein zweistufiger Überfrequenzschutz gefordert. Dieser wird ebenfalls mit vom üEKS abgedeckt, mit dem Frequenzsteigerungsschutz welcher mit den Einstellwerten $f >$ und $f >>$ arbeitet.

Die Einstellwerte der Schutzeinrichtungen werden vom Versorgungsnetzbetreiber mit dem Blatt E.9 „Netzbetreiber-Abfragebogen“, aus der VDE-AR-N 4110 vorgegeben. Damit der Netzbetreiber-Abfragebogen von der WEMAG ausgefüllt werden kann, wird das in der Abbildung 1 geforderte Blatt E.8 aus der VDE-AR-N 4110 benötigt. In dem Blatt E.8 „Datenblatt einer Erzeugungsanlage – Mittelspannung“ werden sämtliche Informationen zu allen hinter dem NAP befindlichen Erzeugungsanlagen abgefragt. Die Vorlagen der Blätter E.8 und E.9 sind den Anhängen A und B zu finden [18]. Zusätzlich fordert die WEMAG ein Single-Line-Diagram (SLD), welches die vollständige Anschlusssituation der Erzeugungsanlagen im Bestand darstellt. In dem SLD sind sämtliche Schutzeinrichtungen eingezeichnet und aufgezeigt, worauf diese wirken. Das SLD des begleiteten Baus ist in der Abbildung 4 abgebildet und ist in voller Größe im Anhang C zu finden.

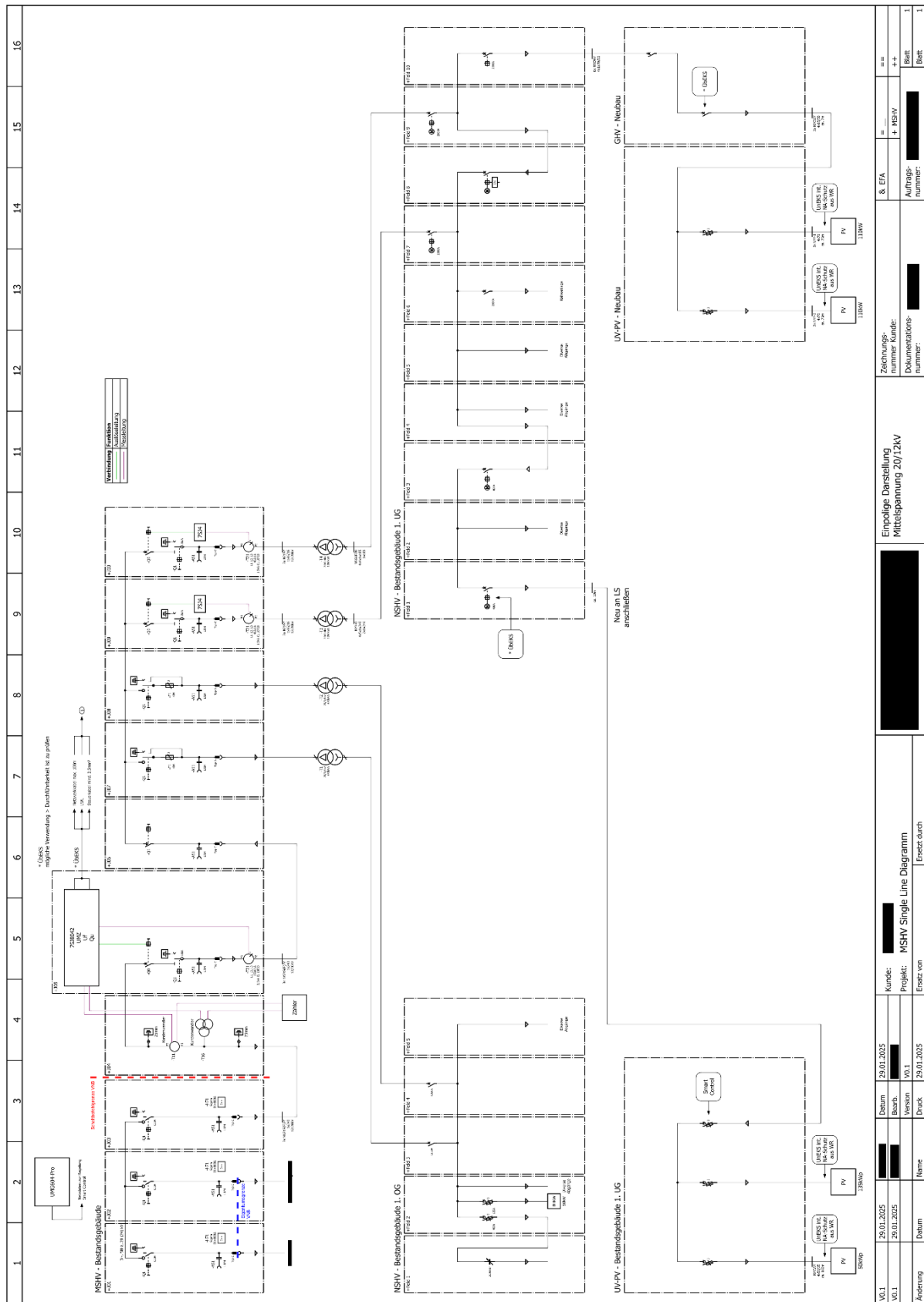


Abbildung 4: Single-Line-Diagramm des begleiteten Baus

Mit den Änderungen durch das Solarpaket 1 entfällt die Pflicht, ein Anlagenzertifikat für die Inbetriebnahme der PV-Anlage vorzulegen. Durch das Solarpaket 1 wurde die Grenze, bei der Anlagenzertifikate erforderlich werden, von einer Anschlusswirkleistung $P_{AV,E}$ von 135 kW auf eine Anschlusswirkleistung $P_{AV,E}$ von 270 kW bzw. einer installierten Leistung $P_{E_{max}}$ von mehr als 500 kW angehoben [19] [20]. Diese Änderungen wurden in der Elektrotechnische-Eigenschaften-Nachweis-Verordnung (NELEV) in §2 Absatz 4 umgesetzt [19]. PV-Anlagen unterhalb dieser Grenzen benötigen lediglich den Nachweis über Einheitszertifikate. Für die im Bau befindliche PV-Anlage ist somit ein Nachweis über Einheitszertifikate ausreichend.

Aus dem Abschnitt 5.7.4.2.1 der VDE-AR-N 4105 geht eine weitere, wesentliche Anforderung hervor. Es muss ein Netzsicherheitsmanagement zur Vermeidung von Netzüberlastungen implementiert werden. Abhängig von der Leistung der PV-Anlage werden unterschiedliche Anforderungen an das Netzsicherheitsmanagement gestellt. Die höchsten Anforderungen werden an PV-Anlagen mit über 100 kWp installierter Leistung gestellt [11]. Die im Bau befindliche PV-Anlage mit 250 kWp fällt in diesen Anforderungsbereich. Daher muss ein Netzsicherheitsmanagementsystem implementiert werden.

Für das Netzsicherheitsmanagement muss eine technische Einrichtung installiert werden, mit der eine ferngesteuerte Leistungsreduzierung der Einspeiseleistung durch den Netzbetreiber vorgenommen und die IST-Einspeisung abgerufen werden kann. Zur Umsetzung des Netzsicherheitsmanagements gibt es seitens der WEMAG die Richtlinie RL N 04 „Technische Umsetzung des Einspeisemanagements bei der WEMAG Netz GmbH“ [21] nach der das Netzsicherheitsmanagement umzusetzen ist.

Für das Netzsicherheitsmanagement gibt es zwei unterschiedliche Steuerungen. Für Anlagen mit einer installierten Leistung von unter 100 kW wird ein Funkrundsteuerempfänger (FRSE) benötigt. Ab einer installierten Leistung über 100 kW muss eine Fernwirkanlage (FWA) verwendet werden [21]. Bei dem begleiteten Bau muss demnach eine FWA verwendet werden. FRSE finden auf Grund ihres Einsatzes bei einer installierten Leistung von

unter 100 kW kaum Anwendung in der Industrie und werden aus diesem Grund nicht weiter betrachtet.

Die bestehenden Erzeugungsanlagen des begleiteten Projekts sind bereits an eine FWA angeschlossen. Diese FWA muss für die neue PV-Anlage erweitert werden, sodass diese über den zu entwickelnden PV-Controller an die FWA angeschlossen werden kann. Die Fernwirkanlage besteht aus einem GPRS-Modem und einer von der WEMAG zugelassenen Fernwerkstation. Das GPRS-Modem wird von der WEMAG zur Verfügung gestellt und übernimmt die sichere Kommunikation mit der Leittechnik der WEMAG via VPN [21]. Aus Abschnitt 6.3.3 der TAB MS ergibt sich ferner die Anforderung an das GPRS-Modem, dieses für mindestens 12 Stunden beim Ausfall des Versorgungsnetzes weiterzubetreiben. Dafür ist eine unterbrechungsfreie Stromversorgung (USV) mit 24 V DC vorzusehen [16]. Die Fernwerkstation wird nicht vom Versorgungsnetzbetreiber gestellt und ist kundenseitig zu realisieren. In der Richtlinie RL N 04 sind im Anhang B zugelassene Fernwerkstationen aufgeführt [21]. Auf Grund der fehlenden Zulassung kann der PV-Controller nicht als Fernwerkstation verwendet werden. Die bestehende FWA nutzt eine Speicherprogrammierbare Steuerung (SPS) der Firma WAGO vom Typen PFC200 als Fernwerkstation.

Im Folgenden wird herausgearbeitet, welche Daten mit der FWA ausgetauscht werden. Die Steuersignale, die mit der WEMAG ausgetauscht werden, sind in der Tabelle 2 in der Datenpunktliste der FWA dargestellt.

Tabelle 2: Datenpunktliste der Fernwirkanlage (Quelle: [21], Seite 10)

Information		Übertragungsadressierung					
		APDU		Info.-Adr.			Bemerkung
		H-Byte	L-Byte	H-Byte	M-Byte	L-Byte	
	Leistungssteuerung (bezogen auf die installierte Nennleistung)						Befehle
<TK45>	Einspeisung 100%	zz	yy	xx	10	1	
<TK45>	Einspeisung 60%	zz	yy	xx	10	3	
<TK45>	Einspeisung 30%	zz	yy	xx	10	5	
<TK45>	Einspeisung 0%	zz	yy	xx	10	7	
							Rückmeldungen
<TK30>	Einspeisung 100%	zz	yy	xx	10	2	
<TK30>	Einspeisung 60%	zz	yy	xx	10	4	
<TK30>	Einspeisung 30%	zz	yy	xx	10	6	
<TK30>	Einspeisung 0%	zz	yy	xx	10	8	
	Schaltgeräte						
<TK46>	LS (Q0) oder LaTS	zz	yy	zz	1	1	Nur Not-Aus
<TK46>	Q (P) Steuerung	zz	yy	zz	1	9	Kennlinie / NLS WEMAG
							Rückmeldungen
<TK31>	LS (Q0) oder LaTS	zz	yy	zz	1	2	
<TK31>	Q (P) Steuerung	zz	yy	zz	1	10	Kennlinie / NLS WEMAG
	Q (P) Steuerung (bezogen auf die installierte Leistung in %)						Sollwertstellbefehl
<TK50>	Blindleistungssteuerung	zz	yy	xx	20	11	
	Messwerte						
<TK13>	IL2	zz	yy	zz	50	1	
<TK13>	UL1_L3	zz	yy	zz	50	2	
<TK13>	P	zz	yy	zz	50	3	Einspeisung in WEMAG Netz = +
<TK13>	Q	zz	yy	zz	50	4	Bezug aus WEMAG Netz = -
<TK13>	cos phi	zz	yy	zz	50	5	induktiv = - / kapazitiv = +
<TK13>	Q / P installiert in %	zz	yy	zz	50	7	untererregt = - / übererregt = +
<TK13>	Windrichtung	zz	yy	zz	50	8	
<TK13>	Windgeschwindigkeit	zz	yy	zz	50	9	
<TK13>	Globalstrahlung	zz	yy	zz	50	10	

Von der Netzleitstelle (NLS) können Steuerbefehle zur Leistungsreduzierung und zur Blindleistungssteuerung vorgegeben werden. Der Befehl zu Leistungsreduzierung wird in Prozent bezogen auf die installierte Nennleistung übermittelt. Für die Blindleistungssteuerung wird vorgegeben, welche Art der Blindleistungssteuerung verwendet wird. Dabei kann zwischen einem Kennlinienbetrieb und der Blindleistungsvorgabe in Prozent bezogen auf die installierte Leistung durch die NLS umgeschaltet werden. Die

Blindleistungskennlinie wird vom Versorgungsnetzbetreiber in dem Blatt E.9 aus der VDE-AR-N 4110 vorgegeben. Dieser Teil des Blatt E.9 ist in der Abbildung 5 dargestellt. Bei der Steuerung der Blindleistung sind abrupte Blindleistungsänderungen zu vermeiden. Als Einregelzeit sind 30 Sekunden pro Blindleistungsstufe einzuhalten [21]. Diese Anforderung wird von den Wechselrichtern selbstständig erfüllt, diese Anforderungen werden dem Wechselrichter über den Länderdatensatz bereitgestellt. Es ist bei der Kommissionierung der Wechselrichter darauf zu achten, den gültigen Länderdatensatz für die VDE-AR-N 4110 auszuwählen [14].

VDE-AR-N 4110:2018-11

Datenabfragebogen Netzbetreiber für Neuanlagen		5 (7)																																					
Anschluss/Änderung einer Erzeugungsanlage																																							
3. Statische Spannungshaltung																																							
Blindleistungsstellbereich	<input type="checkbox"/> 0,95 untererregt bis 0,95 übererregt nach VDE-AR-N 4110 <input type="checkbox"/> untererregt bis übererregt (gesonderte Regelung)																																						
Blindleistungssollwert und Verfahren	<input type="checkbox"/> den TAB vom zu entnehmen																																						
<input type="checkbox"/> Blindleistungs-Spannungs-Kennlinie $Q(U)^9$	Steigung der Kennlinie: Obere Spannungsgrenze $U_{MAX}/U_C =$ (z. B. 1,04) Untere Spannungsgrenze $U_{MIN}/U_C =$ (z. B. 0,96) Maximale Blindleistung $Q_{MAX\text{-}untererregt}/P_{b\text{ inst}} =$ (z. B. 0,33) Spannungstotband = \pm % U_C (z. B. $\pm 1,0$ % U_C) Referenzspannung: <input type="checkbox"/> $U_{Q0,ref}/U_C =$ (z. B. 1,00) <input type="checkbox"/> variabel per Fernwirkanlage ¹⁰																																						
<input type="checkbox"/> Kennlinie $Q(P)^{11}$	$P/P_{b\text{ inst}}$ [%] $Q/P_{b\text{ inst}}$ [%]	<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																					
<input type="checkbox"/> Blindleistung Q mit Spannungsbegrenzungsfunktion	Kennlinie mit $P1 (U_{P1}/U_C; Q_{P1}/P_{b\text{ inst}}) =$ (z. B. 0,94; -0,33) $P2 (U_{P2}/U_C; Q_{ref}/P_{b\text{ inst}}) =$ (z. B. 0,96; 0) $P3 (U_{P3}/U_C; Q_{ref}/P_{b\text{ inst}}) =$ (z. B. 1,04; 0) $P4 (U_{P4}/U_C; Q_{P4}/P_{b\text{ inst}}) =$ (z. B. 1,06; +0,33) <input type="checkbox"/> variabel per Fernwirkanlage ¹⁰ <input type="checkbox"/> Fahrplan ¹²																																						
<input type="checkbox"/> Verschiebungsfaktor $\cos \varphi$	<input type="checkbox"/> $\cos \varphi =$ <input type="checkbox"/> übererregt <input type="checkbox"/> untererregt <input type="checkbox"/> variabel per Fernwirkanlage ¹⁰ <input type="checkbox"/> Fahrplan ¹²																																						
Regelverhalten bei Sollwertsprüngen	Für $Q(U)$, $Q(P)$, Q Zeitkonstante 3 Tau = s (Einstellbereich 10-60 s (Typ 1), 6-60 s (Typ 2))																																						

Abbildung 5: Ausschnitt Blindleistungssteuerung Blatt E.9 der VDE-AR-N 4110 (Quelle: [18], Seite 23)

Von der NLS wird auch ein Not-Aus-Signal übermittelt. Wenn dieses übertragen wird, sind die Erzeugungsanlagen vom Netz zu trennen. Das Not-Aus-Signal wird direkt von der FWA an die LaTS der Erzeugungsanlagen übertragen und schaltet diese frei.

An die NLS werden von der FWA folgende Daten zurückgemeldet: der Not-Aus-Status, die verwendete Blindleistungssteuerung, die aktuelle Leistungsreduzierung, der Strom der

Phase L2, die Leiter-Leiter-Spannung zwischen L1 und L2, die Ist-Einspeisung, die Ist-Blindleistungseinspeisung, die Phasenverschiebung, Q/P_{inst} in % und die Globalstrahlung. Die Windrichtung und Windgeschwindigkeit muss nicht an die WEMAG übertragen werden. Diese Angaben werden nur bei Windkraftanlagen benötigt [21].

In der Abbildung 6 ist der Datenfluss der Fernwirkanlage dargestellt. Die Netzleitstelle überträgt via VPN die Steuersignale an die Fernwirkanlage. Diese Steuersignale werden an den PV-Controller übertragen und weiterverarbeitet. Anschließend werden die Wechselrichter je nach Vorgabe der NLS geregelt. Der PV-Controller meldet die Umsetzung der Steuerbefehle an die FWA zurück. Dabei muss die aktuelle Leistungsreduzierung, die verwendete Blindleistungssteuerung und der Not-Aus-Status zurückgemeldet werden. Zur Erfassung der Globalstrahlung wird ein Pyranometer benötigt. In dem begleiteten Projekt wird das Pyranometer sun[e] 00.16130.501030 des Herstellers Lambrecht meteo verwendet [22]. Das angeschlossene Pyranometer wird durch den PV-Controller wiederholend abgefragt und die Globalstrahlung wird an die FWA übertragen.

Der Stromzähler, der den Netzbezug am NAP misst, kann direkt an die FWA angeschlossen werden. Es ist jedoch auch möglich, den Stromzähler direkt an den PV-Controller anzuschließen. Dann werden die abgefragten Daten des Stromzählers von dem PV-Controller an die FWA übertragen. In dem begleiteten Projekt ist der Stromzähler bereits an die bestehende FWA angeschlossen. Es wird jedoch bei der Erprobung des PV-Controllers angenommen, dass der Stromzähler an dem PV-Controller angeschlossen ist. Die gesammelten Daten werden von der Fernwirkanlage zurück an die Netzleitstelle übertragen. Die Steuersignale der WEMAG sind eigenverantwortlich von dem Anlagenbetreiber nach spätestens 60 Sekunden korrekt umzusetzen [21].

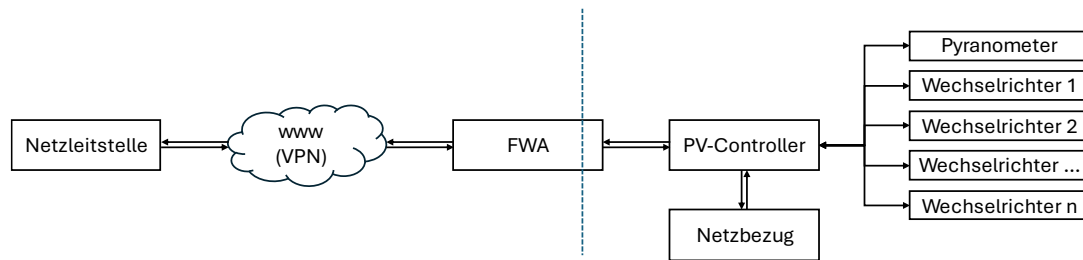


Abbildung 6: Schematische Darstellung Datenfluss der Fernwirkanlage

Aus der Anschlusssituation ergeben sich für den PV-Controller folgende Anforderungen:

- Die Steuerbefehle der FWA müssen von dem PV-Controller abgefragt werden und anschließend korrekt umgesetzt werden. Das Not-Aus-Signal wird dabei direkt von der FWA auf die LaTS übertragen und muss nicht umgesetzt werden.
- Die Umsetzung der Steuerbefehle muss der FWA zurückgemeldet werden.
- Zusätzlich muss der PV-Controller Stromzähler direkt abfragen können, um auch bei einer anderen Bestandssituation einsetzbar zu sein.
- Zur Messung der Globalstrahlung muss der PV-Controller ein Pyranometer abfragen können.
- Die Steuersignale des Netzbetreibers müssen dabei innerhalb von 60 Sekunden umgesetzt sein.
- Um einer Abschaltung der Erzeugungsanlagen durch die $P_{AV,E}$ -Überwachung vorzugreifen, wird die PV-Anlage beim Erreichen einer konfigurierbaren Leistungsgrenze frühzeitig gedrosselt, um einer Abschaltung vorzugreifen und die Nutzung des selbst erzeugten Stroms zu maximieren.

Die weiteren geforderten Schutzeinrichtungen arbeiten autark und stellen keine weiteren Anforderungen an den PV-Controller. Aus den TABs sind keine weiteren Anforderungen an den PV-Controller entstanden.

2.4 Steuerbare Verbrauchseinrichtungen

Bei einem hohen Netzbezug kann zur Reduzierung der Spitzenlast die Leistungsaufnahme von steuerbaren Verbrauchseinrichtungen reduziert werden. Eine solche Leistungsreduzierung kann aus Sicht des Netzes sinnvoll sein. Es können jedoch unter anderem wirtschaftliche Gründe dagegensprechen. Beispielsweise wenn ein Elektroauto geladen wird, dessen Ladevorgang Einnahmen generiert, ist eine Leistungsreduzierung dieses Ladevorgangs ggf. unwirtschaftlich. Eine Leistungsreduzierung an anderer Stelle wäre in diesem Fall geeigneter.

Auf Grund dessen, dass der PV-Controller die aktuelle Situation nur innerhalb seiner Systemgrenzen betrachten kann und die Information über die Gesamtsituation nicht vorliegt, kann aus Sicht des PV-Controllers eine Leistungsreduzierung sinnvoll sein. Diese kann jedoch unter Berücksichtigung der Gesamtsituation von Nachteil sein. Daher wurde sich dafür entschieden, den steuerbaren Verbrauchern eine Empfehlung zu geben, ob eine Leistungsreduzierung aus Sicht des Netzes sinnvoll ist, etwa zur Reduzierung der Spitzenlast. Durch Reduzierung der Spitzenlast wird das zu zahlende Netzentgelt reduziert. Ausschlaggebend für das Netzentgelt bei der Abnahme von mehr als 100.000 kWh/a ist die höchste auf 15 Minuten gemittelte Spitzenlast [23]. Durch Reduzierung dieser kann das zu zahlende Netzentgelt deutlich verringert werden. Die Empfehlung zur Leistungsreduzierung wird in binärer Form ausgegeben. Eine 0 wird ausgegeben, falls keine Leistungsreduzierung vorgenommen werden soll und eine 1, wenn eine Leistungsreduzierung sinnvoll erscheint. Das Empfehlungssignal wird auf dem PV-Controller bereitgestellt. Bei einer zukünftigen Erweiterung des PV-Controllers um die Steuerung von Verbrauchern kann die ausgegebene Empfehlung zur Leistungsreduzierung direkt auf dem PV-Controller weiterverarbeitet werden. Das Empfehlungssignal könnte dabei auch über das Automatisierungsprotokoll OPC UA anderen Controllern bereitgestellt werden. Ob und in welche Größe eine Leistungsreduzierung vorgenommen wird, kann somit einem übergeordneten Controller, z.B. einem Lademanagement-Controller einer Ladeinfrastruktur überlassen werden.

3 Grundkonzept

Anhand der in der Anforderungsanalyse herausgearbeiteten Anforderungen an den PV-Controller wird in diesem Abschnitt das Grundkonzept für die Programmierung erarbeitet. Dafür wird zunächst die Programmierumgebung ausgewählt. Anschließend werden die erforderlichen Schnittstellen des RasPi sowie ein industrietaugliches RasPi Modell ausgewählt. Abschließend werden die für die Modbus-Kommunikation benötigten Daten ermittelt und die Konfigurierbarkeit der Programmierumgebung herausgearbeitet.

3.1 Auswahl der Programmierumgebung

Bei der Auswahl der verwendeten Programmiersprache soll der Open-Source-Gedanke, den auch die Energie-Monitoring-Software JEVIs verfolgt, weitergeführt werden. Des Weiteren muss die Programmiersprache plattformübergreifend einsetzbar sein, um nicht an eine spezielle Plattform gebunden zu sein. So wird sichergestellt, dass auch bei etwaigen Preisanstiegen einzelner Plattformen, der PV-Controller kostengünstiger auf einer anderen Plattform implementiert werden kann.

Die Programmierumgebung Node-RED passt sehr gut zu den gestellten Anforderungen. Node-RED wird seit Anfang 2013 entwickelt und wurde bereits im September 2013 unter eine Open-Source-Lizenz gestellt. Seitdem wurde Node-RED kontinuierlich weiterentwickelt [24]. Durch die Open-Source-Lizenz ist es der Node-RED-Community möglich, eigene Nodes, Bausteine für die flussbasierte Programmierung, in Node-RED zu erstellen und sie der Community kostenlos als Paletten zur Verfügung zu stellen. Eine Palette ist eine Sammlung zusammengehöriger Nodes. Diese Paletten erweitern Node-RED um weitere Funktionen. In einem früheren Projekt wurden von Envidatec bereits mehrere Paletten zur Erfassung von Energiedaten für die Energie-Monitoring-Software JEVIs erstellt [25] [26]. Diese Paletten können bei der Programmierung des PV-Controllers verwendet werden. Dies hat zusätzlich den Vorteil, dass ein Mitarbeiter der Envidatec bereits Erfahrung

mit Node-RED sammeln konnte und der PV-Controller in Zukunft durch mehrere Mitarbeiter weiterentwickelt werden kann.

In Node-RED können unter Zuhilfenahme von Function-Nodes schnell und unkompliziert in JavaScript eigene Funktionen programmiert werden. Node-RED selbst basiert ebenfalls auf JavaScript [27].

Node-RED hat in den vergangenen Jahren einen erheblichen Zulauf im Bereich des industriellen Internets der Dinge gewonnen (engl.: Internet of Things, kurz: IoT). So wird Node-RED von unterschiedlichen Herstellern von SPS- und IoT-Lösungen als Standardlösung angeboten. Node-RED ist nicht an eine bestimmte Plattform gebunden. So läuft Node-RED auf RasPis oder jedem anderen Debian basierten Betriebssystem sowie auf Windows, Android und Linux Distributionen [28]. Dadurch ist es möglich, Node-RED auf einer Vielzahl unterschiedlicher Controller zu nutzen. Neben dem RasPi läuft Node-RED zum Beispiel auch auf Geräten der Firmen WAGO [29] und Siemens [30]. Dadurch ist es möglich, den PV-Controller auf eine andere Plattform zu migrieren.

Node-RED wird immer öfter als Standardlösung angeboten und wird von einer Vielzahl an Geräten unterstützt. Des Weiteren wurden von der Envidatec bereits Palette für Node-RED entwickelt, diese können bei der Programmierung genutzt werden. Aus diesen Gründen wurde sich für Node-RED als Programmierumgebung entschieden.

Zur Verbesserung der Lesbarkeit werden die Namen der Nodes oder anderer Inhalte von Node-RED, die im Text erwähnt werden, in der Schriftart `Courier New` dargestellt.

Zum besseren Verständnis von Node-RED wird nachfolgend erläutert, in welcher Form die Daten zwischen den Nodes übertragen werden. Node-RED arbeitet mit `message objects` (`msg.object`). Diese werden von den Nodes am Eingang eingelesen und nach der Weiterverarbeitung am Ausgang des Nodes wieder ausgegeben. Innerhalb dieses Objektes können beliebig viele Informationen gespeichert werden. In dem `msg.payload` Objekt werden typischerweise die Informationen gespeichert, die die Nodes zum Arbeiten benötigen. Diese Informationen können auch an anderer Stelle im `msg.object`

gespeichert werden. In dem in Abbildung 7 gezeigten Beispiel `msg.object` wird der Dateipfad, der von einem Node benötigt wird, unter `msg.filename` gespeichert und nicht im `payload` des `msg.object`. Jedes `msg.object` besitzt zudem eine individuelle `msgid` (Message ID), mit der eine eindeutige Zuordnung vorgenommen werden kann.

```
1.2.2025, 12:51:11 node: debug 143
msg : Object
  ▾ object
    _msgid: "c8fc2eae856c1367"
    ▾ payload: object
      col1: "Solar control"
      col2: "Config"
      col3: "Max grid consumption"
      col4: "1200"
      filename:
        "/home/pi/Konfiguration.csv"
      columns: "col1,col2,col3,col4"
    ▸ parts: object
```

Abbildung 7: Beispiel `msg.object`

Für die Speicherung der Daten im `msg.object` können alle in JavaScript unterstützten Datentypen verwendet werden. So beispielsweise Boolean, Number, Array, String, Object und Null [31]. Der Datentyp wird dabei selbständig von Node-RED erkannt.

3.2 Auswahl der erforderlichen Schnittstellen

Bevor ein geeigneter industrietauglicher RasPi für die Implementierung ausgewählt werden kann, müssen die erforderlichen Schnittstellen definiert werden. In den Design-Vorgaben des Unternehmens (s. Abschnitt 2.2) wird vorgegeben, dass nach Möglichkeit eine Modbus TCP Verbindung für die angeschlossenen Geräte verwendet werden soll. Modbus RTU Verbindungen werden mit einem RS-485 zu TCP/IP Umsetzer in eine Modbus TCP Verbindung umgewandelt. Da die Anforderungen an den PV-Controller anhand des begleiteten Projekts abgeleitet werden, muss darauf geachtet werden, dass alle

verwendeten Geräte an den PV-Controller angebunden werden können. Im Folgenden wird für die anzuschließenden Geräte überprüft, ob diese Modbus TCP unterstützen.

Bei dem Projekt werden die Wechselrichter Sunny Tripower CORE 2 vom Hersteller SMA verwendet. Aus dem Datenblatt geht hervor, dass der Wechselrichter die Datenschnittstellen Webinterface und Modbus SunSpec besitzt [32]. Über das Webinterface werden die Daten des Wechselrichters in das Webportal Sunny Portal von SMA übertragen. Bei der Schnittstelle Modbus SunSpec wird das reguläre Modbus TCP Protokoll verwendet. Die WAGO SPS vom Typen PFC200 welche als Fernwerkstation in der FWA eingesetzt wird, bietet ebenfalls die Möglichkeit zur Kommunikation über Modbus TCP [33]. Als Stromzähler zur Messung des Netzbezuges wird ein Janitza Measuring Instruments Directive (MID) Energiezähler B24 mit Kabelumbauwandlern verwendet. Dieser unterstützt nur eine Modbus RTU Verbindung [34]. Zur Anbindung dieses Gerätes über Modbus TCP wird ein RS-485 zu TCP/IP Umsetzer verwendet. Gleiches gilt auch für das verwendete Pyranometer sun[e] 00.16130.501030 von Lambrecht meteo [22]. Dieses Gerät wird ebenfalls über einen ein RS-485 zu TCP/IP Umsetzer in eine Modbus TCP Verbindung umgewandelt.

Alle der verwendeten Geräte können über Modbus TCP an den PV-Controller angeschlossen werden. Demnach ist es ausreichend, wenn der verwendete RasPi eine Modbus TCP Verbindung unterstützt bzw. einen Ethernet Anschluss besitzt. Dieser lässt sich durch Node-RED für eine Modbus TCP Verbindung nutzen.

Eine Vielzahl von Wallboxen, z.B. KEBA und Compleo unterstützen Modbus TCP bzw. Modbus RTU [35], [36]. Dadurch ist es bei einer zukünftigen Erweiterung des PV-Controllers möglich Ladeinfrastruktur zu unterstützen, ohne das weitere Anforderungen an die Schnittstellen entstehen.

3.3 Auswahl des Raspberry Pi

Für die Implementierung des PV-Controllers wurde durch die Envidatec ein RasPi vorgegeben. Die Auswahl eines geeigneten RasPi, welcher für die Implementierung verwendet wird, wird im Folgendem getroffen.

Der RasPi wird in einem Schaltschrank verbaut, dafür muss er auf einer Hutschiene montierbar sein. Er muss ebenfalls mit der industrietypischen Versorgungsspannung von 24 V DC betrieben werden können.

Als Hersteller von Industrie Raspberry Pi, welche diese Anforderungen erfüllen, gibt es u.a. die Firmen KUNBUS und SFERALABS [37], [38]. Die Geräte von KUNBUS haben gegenüber den SFERALABS Geräten den Vorteil, dass Sie die Norm EN/IEC 61131-2 vollständig erfüllen [39]. Diese Norm befasst sich mit den Betriebsmittelanforderungen und Prüfungen von Speicherprogrammierbaren Steuerungen, um sicherzustellen, dass diese in industriellen Umgebungen zuverlässig funktionieren. Die Geräte des Herstellers SFERALABS haben keine EN/IEC 61131-2 Zertifizierung. Aus diesem Grund werden im Folgenden nur noch die Geräte des Herstellers KUNBUS betrachtet, um eine zuverlässige Funktion des PV-Controllers zu gewährleisten. Von KUNBUS stehen mehrere Geräte der Produktfamilie Revolution Pi (RevPi) zur Auswahl, diese sind in der Tabelle 3 dargestellt.

Tabelle 3: RevPi Modelle und Preise [40]

Modell	Listenpreis (netto)
KUNBUS RevPi Connect S	ab 407 €
KUNBUS RevPi Connect SE	ab 377 €
KUNBUS RevPi Connect 4	ab 389 €
KUNBUS RevPi Connect 5	ab 499 €
KUNBUS RevPi Core S	ab 278 €
KUNBUS RevPi Core SE	ab 247 €
KUNBUS RevPi Flat S	ab 396 €
KUNBUS RevPi Compact	ab 481 €

Diese Geräte erfüllen alle die an die Schnittstellen gestellten Anforderung nach einem Ethernet Anschluss, über dem mit Modbus TCP kommuniziert werden kann. Die Geräte unterscheiden sich im Wesentlichen durch die unterschiedlichen Erweiterungsmodule mit denen der jeweilige RevPi erweitert werden kann. Der KUNBUS RevPi Flat S und der RevPi Compact können keine Erweiterungsmodule aufnehmen. Die verfügbaren Erweiterungsmodule für die verbleibenden Geräte sind in der nachfolgenden Tabelle 4 abgebildet.

Tabelle 4: Übersicht RevPi Erweiterungsmodule [40]

Erweiterungsmodule	Kurzübersicht
RevPi DIO	- 14 digitale Ein- und Ausgänge
RevPi DI	- 16 digitale Eingänge
RevPi DO	- 16 digitale Ausgänge
RevPi AIO	- 4 analoge Eingänge - 4 analoge Ausgänge - 2 Temperaturkanäle
RevPi MIO	- 8 analoge Eingänge - 8 analoge Ausgänge - 4 digitale Eingänge - 4 digitale Ausgänge
RevPi RO	- 4 Relais Output
RevPi Con M-Bus	- 868 MHz Wireless M-Bus Interface - Nur RevPi Connect S und SE.
RevPi Con M-Bus VHP	- 169 MHz Wireless M-Bus Interface - Nur RevPi Connect S und SE.

Erweiterungsmodule	Kurzübersicht
RevPi Con CAN	<ul style="list-style-type: none"> - CAN-Bus Interface - Nur RevPi Connect S und SE.
RevPi Gate PROFINET	<ul style="list-style-type: none"> - PROFINET Interface (nur Slave) - Nur RevPi Connect 5, RevPi Connect S und RevPi Core S.
RevPi Gate EtherNet/IP	<ul style="list-style-type: none"> - EtherNet/IP interface (nur Slave) - Nur RevPi Connect 5, RevPi Connect S und RevPi Core S.
RevPi Gate EtherCAT	<ul style="list-style-type: none"> - EtherCAT interface (nur Slave) - Nur RevPi Connect 5, RevPi Connect S und RevPi Core S.
RevPi Gate PROFIBUS	<ul style="list-style-type: none"> - PROFIBUS interface (nur Slave) - Nur RevPi Connect 5, RevPi Connect S und RevPi Core S.

Diese Erweiterungsmodule werden nicht benötigt, da bereits alle Basis-Versionen der RevPi Geräte die Forderung nach einem Ethernet Anschluss erfüllen. Falls der PV-Controller jedoch in Zukunft erweitert werden soll, können diese Erweiterungsmodule genutzt werden. Aus diesen Gründen wird sich für den RevPi Core SE entschieden, da dieser das preiswerteste RevPi Modell ist, welches die gestellten Anforderungen erfüllt [41]. Dieses Modell kann bei Bedarf um digitale Ein- und Ausgänge sowie um Relais-Ausgänge erweitert werden. Falls Gate (Gateway) oder Con (Connection) Erweiterungsmodule in Zukunft benötigt werden kann das RevPi Modell, ohne Änderungen am hier entwickelten PV-Controller, gewechselt werden.

3.4 Benötigte Daten für die Modbus-Kommunikation

Bei der Modbus-Kommunikation sind alle Daten, welche gelesen oder geschrieben werden können, in einem fortlaufenden Speicher gespeichert. Dieser Speicher besteht aus maximal 65.536 Registern mit einer Länge von 16 Bit. Dabei können sich Daten über mehrere Register erstrecken. Deshalb wird für die gerätespezifische Datenzuordnung eine dokumentierte Modbus-Registertabelle benötigt [42]. Modbus-Registertabellen verschiedener Geräte sind nicht einheitlich, so muss für jedes angeschlossene Gerät die jeweils zugehörige Modbus-Registertabelle verwendet werden.

Aus den Modbus-Registertabellen müssen folgende Informationen in die Konfiguration übernommen werden, damit die Datenpunkte korrekt gelesen bzw. geschrieben werden können.

Function Code (FC)

Der FC gibt die Richtung der Übertragung an. Es wird der FC 3 (Read Holding Register) genutzt, wenn ein Register auf dem angeschlossenen Gerät gelesen werden muss. Beim Schreiben eines Registers wird der FC 6 (Write Single Register) genutzt. Weitere FCs werden für die Modbus-Kommunikation nicht benötigt. Die Register können als read only, write only oder read write ausgeführt werden, diese Festlegung wird durch den Hersteller des Gerätes vorgegeben [42].

Adresse

Die Adresse gibt an, in welchem der maximal 65.536 Register der Datenpunkt beginnt [42].

Quantity

Die Quantity gibt an wie viele Register ab der Adresse gelesen bzw. geschrieben werden müssen, um den Datenpunkt vollständig abzufragen bzw. zu schreiben. Abhängig von Datentyp müssen unterschiedlich viele Register gelesen bzw. geschrieben werden. Ein einzelnes Register hat eine Länge von 16 Bit. Wenn ein Float Wert mit einer Länge von 32 Bit gelesen oder geschrieben werden soll müssen demnach 2 Register berücksichtigt werden [42].

Unit-ID

Über die Unit-ID wird bei Modbus RTU Verbindungen verwendet um die an den RS-485 Bus angeschlossenen Geräte zu adressieren. Bei Modbus TCP Verbindungen wird keine Unit-ID benötigt, da die Adressierung durch die IP-Adresse vorgenommen wird. Die Unit-ID wird mit in die Konfiguration aufgenommen, um auch Modbus RTU Geräte welche über einen RS-485 zu TCP/IP Umsetzer, an den an den PV-Controller angeschlossen sind, abfragen zu können [43].

Diese Informationen müssen für jeden Modbus-Datenpunkt in der Konfiguration an den PV-Controller übergeben werden und bei der Modbus-Kommunikation berücksichtigt werden.

3.5 Konfigurierbarkeit von Node-RED

Zur Konfiguration des PV-Controllers wird ein standardisiertes Excel-Tool entwickelt, mit dem die projektspezifische Konfiguration des PV-Controllers vorgenommen werden soll. Dabei sollen möglichst viele Konfigurationen über eine Datei in den PV-Controller eingelesen werden können. Zunächst wird betrachtet, wie eine Datei in Node-RED eingelesen werden kann. Zum Einlesen einer Datei gibt es den `read file Node`. Mit diesem Node ist es möglich, den Dateiinhalt als Zeichenfolge (string) oder als binären Puffer (buffer) auszugeben. Bei der Ausgabe des Dateiinhaltes als Zeichenfolge können die Daten unter

Zuhilfenahme eines `Parser` Nodes weiterverarbeitet werden. Es gibt für die Dateitypen `csv`, `html`, `json`, `xml` und `yaml` Parser in den Standard Nodes.

Es wurde sich für den Dateityp `.csv` entschieden, da dieser Dateityp in der Industrie für den Austausch von Daten weitverbreitet ist. Zudem besitzt dieser Dateityp einen einfachen Aufbau und kann, durch das Anhängen weiteren Datenzeilen und -spalten, einfach erweitert werden.

Mit dem `csv`-Parser ist es möglich, die Daten in einzelnen `msg.object` ausgeben zu lassen und diese anschließend mit einem `Switch` Node anhand ihrer Bezeichner aufzuteilen und durch einen `Change` Node als globale oder Flow Variable in dem jeweiligen Flow zu speichern.

In der Abbildung 8 ist dieses Konzept zum Einlesen der Konfiguration abgebildet. Über `Link` Nodes werden die Flows untereinander verbunden, so ist es möglich die Konfiguration in den zugehörigen Flow zu speichern.

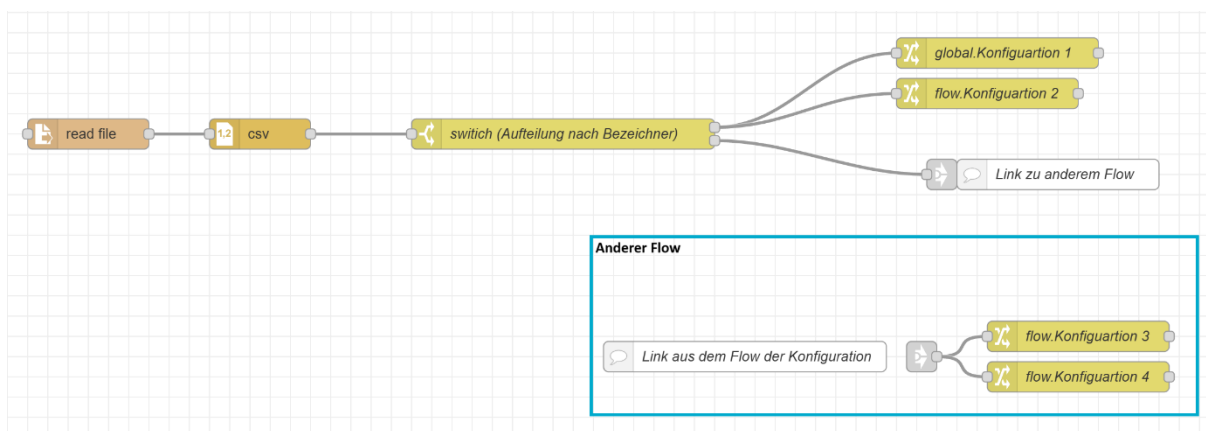


Abbildung 8: Konzept zum Einlesen der Konfiguration in Node-RED

Die gespeicherte Konfiguration kann anschließend auf dem PV-Controller verwendet werden. Für das Einlesen der Konfiguration gibt es eine Einschränkung durch die technischen Möglichkeiten von Node-RED. So ist es während der Laufzeit des Programmes nicht möglich Anpassungen an den Einstellungen der Nodes vorzunehmen. Aus diesem Grund müssen die Einstellungen der `Modbus Flex Getter` Nodes und `Modbus Flex Write`

Nodes, welche für die Modbus-Kommunikation benötigt werden, händisch angepasst werden.

Es müssen folgende Daten angepasst werden:

- Host (IP-Adresse)
- Port

Wie dieser Teil der Konfiguration auf den PV-Controller übertragen wird, ist in dem Excel-Konfigurationstool beschrieben, sowie in dem Abschnitt 6.9 Konfiguration übertragen

3.6 Anforderungen an die Node-RED Programmierung

Node-RED kann durch von der Community erstellte Nodes um weitere Funktionen erweitert werden. Diese Nodes lassen sich als sogenannte Paletten zu Node-RED hinzufügen. Für die Programmierung des PV-Controllers werden weitere Nodes benötigt, welche nicht in den Standard Nodes von Node-RED enthalten sind. Dabei ist darauf zu achten, dass diese Paletten in Zukunft weiter gepflegt werden, um sicherzustellen dass der PV-Controller weiterverwendet werden kann, ohne dass ein Teil der Programmierung erneut vorgenommen werden muss. Daher wird bei der Programmierung darauf geachtet möglichst wenig Paletten zu verwenden. Zudem wird darauf geachtet, dass diese in der Node-RED-Community aktiv gepflegt werden. Die bei der Programmierung des PV-Controllers verwendeten Paletten sind in der Abbildung 9 dargestellt.

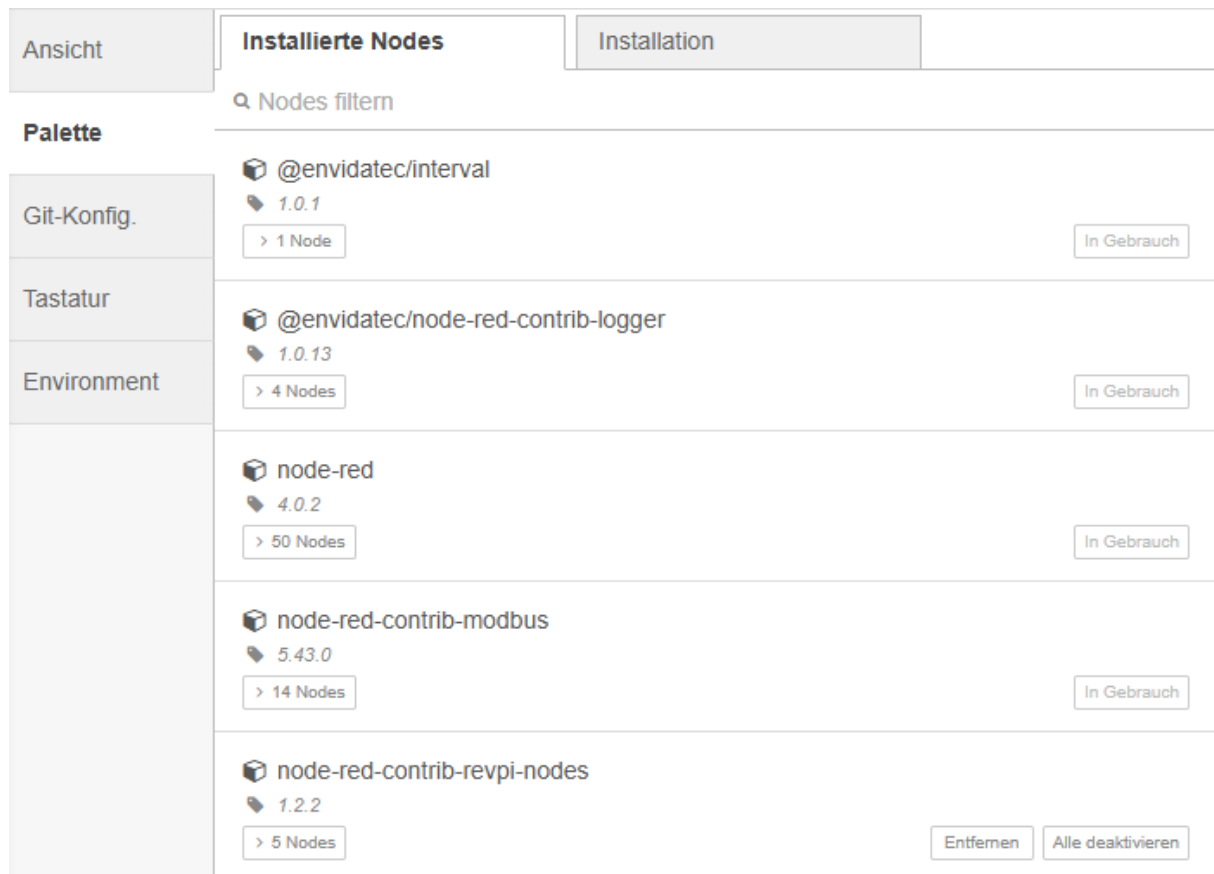


Abbildung 9: Verwendete Paletten für die Node-RED Programmierung

Es wurden zwei von der Envidatec entwickelte Paletten verwendet, diese werden kontinuierlich von der Envidatec für die Interaktion mit der Energie-Monitoring-Software JEVIS gepflegt. Die Palette `@envidatec/interval` enthält einen Intervall Node, welcher zu bestimmten Zeiten ein `msg.object` passieren lässt [25]. Die zweite Palette `@envidatec/node-red-contrib-logger` wird zum Loggen der von den Geräten abgefragten Daten in einer Datenbank benötigt [26]. Neben den von der Envidatec entwickelten Paletten wird noch die Palette `node-red-contrib-modbus` verwendet. Dies ist eine Langzeit unterstützte Palette für die Modbus-Kommunikation. Die Palette wird von der P4NR B2B Community mit Unterstützung durch Iniationware kontinuierlich weiterentwickelt und verbessert [44]. In der Abbildung 9 ist zusätzlich die Palette `node-red-contrib-revpi-nodes` abgebildet. Sie wird jedoch, wie an dem fehlenden Hinweis „Im Gebrauch“ zu erkennen, nicht verwendet und beinhaltet spezielle Nodes für die Verwendung auf

Revolution Pis. Deren Verwendung wäre widersprüchlich zur gewollten Herstellerunabhängigkeit.

4 Programmkonzept

Anhand der ausgearbeiteten Anforderungen und des Grundkonzeptes wird nun das Programmkonzept erstellt. Dafür wird zunächst die Programmstruktur geplant und festgelegt, wie viele Geräte von dem PV-Controller zunächst unterstützt werden. Anschließend wird für eine einfache Distribution der Software und für die sichere Speicherung des Programmierfortschritts ein Git bzw. ein Github aufgesetzt.

4.1 Programmstruktur

Bei der Planung der Programmstruktur liegt ein besonderes Augenmerk darauf, dass das Programm zukünftig erweiterbar ist. Zielsetzung ist, dass der PV-Controller im Anschluss an diese Arbeit weiterentwickelt und für mindestens zehn Jahre unterstützt werden kann.

Die Struktur des Programms muss daher so entworfen werden, dass der PV-Controller um weitere Funktionen erweitert werden kann, ohne größere Anpassungen am bestehenden Programm durchzuführen. Dafür werden die einzelnen Funktionen und Teilbereiche des Programms auf separate `Flows` verteilt. Dadurch wird ermöglicht, dass verschiedene `Flows` auf einzelne Funktionen zugreifen können, so dass gleiche Teilfunktionen nicht mehrfach an unterschiedlichen Stellen im Programm implementiert werden müssen. Auf diese Weise wird eine übersichtliche Programmstruktur geschaffen, welche zum einen leichter zu warten ist und zum anderen die zukünftigen Programmieraufwände, bei gleichbleibender Qualität, reduziert. Zusätzlich muss die Grundstruktur erlauben, dass weitere Geräte für das Monitoring oder für Steuerungsaufgaben hinzugefügt werden können.

Vor dem Beginn der Programmierung wurde in Node-RED ein schematischer Aufbau der Programmstruktur in einem `Flow` erstellt. Dieser führt keine programmtechnischen Funktionen aus, sondern dient der Dokumentation und Veranschaulichung der Software-Implementierung. In Abbildung 10 ist der schematische Aufbau des Programmablaufes dargestellt.

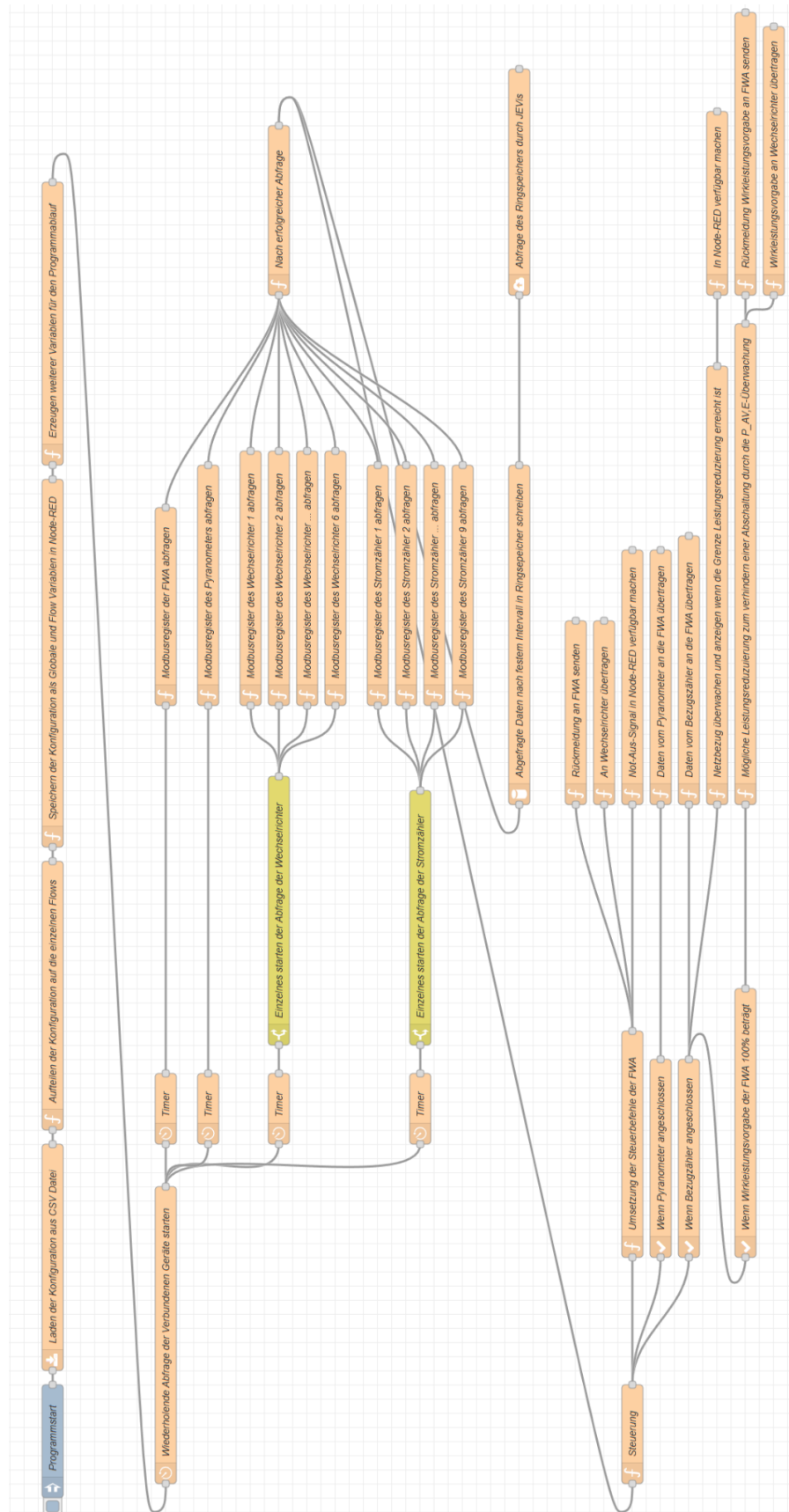


Abbildung 10: Darstellung der Programmstruktur als Flow in Node-RED

Nach dem Start des PV-Controllers, wird automatisch die mit dem Excel-Konfigurations-tool erstellte csv-Datei, mit in die benötigten Konfigurationsparametern, importiert. Beim Einlesen der Konfigurationsdatei werden die Parameter an die jeweiligen `Flows` übergeben und dort als sogenannte Flow-Variablen gespeichert. Auf Flow-Variablen kann nur innerhalb des jeweiligen `Flows` zugegriffen werden. Durch die getrennte Speicherung in den `Flows` wird die Übersichtlichkeit des Programms erhöht. Dies hat zusätzlich den Vorteil, dass `Flows` kopiert werden können, ohne dass Variablennamen angepasst werden müssen. So gibt es z.B. einen `Flow` für jeden Wechselrichter, welcher intern mit denselben Variablennamen arbeiten kann, ohne dass es zu Konflikten kommt. Für die Kommunikation mit der FWA und dem Pyranometer werden entsprechend eigene `Flows` erstellt. Aufgrund dessen, dass für das Energiemonitoring eine Vielzahl an Stromzählern abgefragt werden, sind jeweils drei Stromzähler in einem `Flow` zusammengefasst. Dadurch reduziert sich die Anzahl der benötigten `Flows`.

Einige Konfigurationsparameter werden an mehreren Stellen im Programm benötigt, wie z.B. die Speicherdauer des Ringspeichers der Wechselrichter. Diese Konfigurationen werden als globale Variablen in Node-RED angelegt. Auf globale Variablen kann aus jedem `Flow` des Programms zugegriffen werden. Nach dem Einlesen der Konfiguration werden zusätzlich Merker-Variablen, welche für den Programmablauf benötigt werden, erzeugt. Welche Merker-Variablen benötigt werden und an welcher Stelle im Programm diese erzeugt werden, wird im Abschnitt 5 Programmierung beschrieben.

Nachdem der PV-Controller konfiguriert ist, startet der zyklische Teil des Programmablaufes. Dabei sind die verbundenen Geräte in vier Gruppen unterteilt. Die FWA, das Pyranometer, die Wechselrichter und die Stromzähler bilden jeweils eine Gruppe. Durch Timer wird die Abfrage dieser Gruppen in festgelegten Intervallen ausgeführt. Die Geräte der jeweiligen Gruppe werden sequenziell nacheinander abgefragt. Dadurch wird eine mögliche Überlastung des Modbus verhindert und die Performance des PV-Controller verbessert sich, wodurch eine ausreichende Leistungsreserve für zukünftige Erweiterungen gesichert wird.

Nach der erfolgreichen Abfrage der verbundenen Geräte werden die Daten verarbeitet. Alle abgefragten Daten werden periodisch in dem Ringspeicher des PV-Controllers abgespeichert. Die Energie-Monitoring-Software JEVIS kann den Ringspeicher auslesen und die Daten übergeordnet aufbereiten und weiterverarbeiten. Bei der Abfrage der Daten von den Geräten werden die `Flows`, die die Steuerung übernehmen, automatisch mit den abgefragten Daten aufgerufen. Dabei übernimmt ein `Flow` die Umsetzung der Steuerbefehle der FWA und der Leistungsreduzierung, um einer Abschaltung durch die $P_{AV,E}$ -Überwachung vorzugreifen, vor. Von dem Flow werden die benötigten Steuereingriffe an den Wechselrichter und die Rückmeldung an die FWA, über die Umsetzung der Steuerbefehle, ausgegeben und an den jeweilige Flow zur Übertragung zurückgegeben. Wenn ein Bezugszähler angeschlossen ist, werden die von der FWA benötigten Daten an diese übertragen.

In einem weiteren `Flow` wird der Netzbezug ausgewertet und bei einer Überschreitung des eingestellten Grenzwertes, ab dem eine Lastspitze droht, eine Meldung ausgegeben. Diese Meldung wird zunächst intern auf dem PV-Controller bereitgestellt und könnte in Zukunft über zusätzliche Automationsprotokolle, wie z.B. OPC UA externen Steuergeräten, z.B. für das Lademanagement einer Ladeinfrastruktur, zur Verfügung gestellt. Für den Fall, dass ein Pyranometer angeschlossen ist, wird die Globalstrahlung an die FWA übertragen.

4.2 Unterstützte Geräte

Vor dem Beginn der prototypischen Implementierung wurde die maximale Anzahl der angeschlossenen Geräte sowie die unterstützten Gerätetypen festgelegt. Für jede angeschlossene Gerätearten wird ein exemplarischer Gerätetyp implementiert. In dem Konfigurationstool können zukünftig weitere Modbus-Registerlisten hinterlegt werden, um weitere Geräte zu unterstützen.

Es wurde sich dafür entschieden, zunächst maximal sechs Wechselrichter an einem PV-Controller zu unterstützen. Die beim begleiteten Projekt verwendeten Wechselrichter

SUNNY TRIPOWER CORE 2 (STP 110-60) der Firma SMA verwenden den Modbus-Datenstandard SunSpec [45] [46]. SunSpec wird von der SunSpec Alliance definiert und verfolgt das Ziel, den Ausbau dezentraler Energiewirtschaft zu beschleunigen, indem die Modbus-Kommunikation durch eine herstellerübergreifende Standardisierung vereinfacht wird. Hersteller können ihre Geräte nach Implementierung des Datenstandards durch die SunSpec Alliance zertifizieren lassen, so dass die Interoperabilität verschiedener Geräte sichergestellt wird. Bei dem PV-Controller wird die SunSpec Modbus-Kommunikation mit dem SMA SUNNY TRIPOWER CORE 2 implementiert. Weitere Wechselrichter, die SunSpec Modbus unterstützen, können durch das Hinterlegen weiterer Modbus-Registerlisten leicht implementiert werden. Die Register des SunSpec Modbus sind vordefiniert sowie der Offset der Register zueinander, nur die Startadressen unterscheiden sich voneinander [47] [48]. Zum Hinzufügen weiterer Wechselrichter mit SunSpec Modbus Unterstützung müssen demnach nur die Adressen in der vorhandenen Modbus-Registerliste angepasst werden und in das Konfigurationstool eingebunden werden. Mögliche Hersteller weiterer Wechselrichter sind unter anderem Huawei, ABB und Sungrow [49].

Die maximale Anzahl der unterstützten Stromzähler an einem PV-Controller wurde zunächst auf neun festgelegt. Als Stromzähler wird der Janitza MID-Energiezähler B24 implementiert. Dieser Zähler wird in dem begleiteten Projekt verwendet. Die maximale Anzahl von Stromzählern wird in Zukunft weiter erhöht, wenn der PV-Controller verstärkt auch für das Energiemonitoring eingesetzt wird. Aus kaufmännischen Gründen wären 50 Stromzähler, die angeschlossen werden können, denkbar. So können bei der Unterstützung von weniger Geräten mehrere PV-Controller benötigt werden. Dadurch erhöht sich die Gewinnmarge und es kann mehr Hardware verkauft werden. Die finale Anzahl sollte sich an den auf dem Markt befindlichen vergleichbaren Controllern orientieren und in Zukunft festgelegt werden. Ebenfalls können weitere Stromzählermodelle durch das Hinterlegen weiterer Modbus-Registerlisten in dem Konfigurationstool hinzugefügt werden. Dabei kann in der Konfiguration ausgewählt werden, ob es sich bei einem der Stromzähler um den Bezugzähler handelt.

Da eine FWA unabhängig von der Anzahl der Erzeugungsanlagen benötigt wird, wird nur eine FWA von dem PV-Controller unterstützt. Auf der WAGO SPS PFC200, die in dem begleiteten Projekt in der FWA als Fernwirkstation eingesetzt wird, werden die Steuersignale aus der Tabelle 2 der WEMAG über Modbus bereitgestellt. Auf Grund der individuellen Programmierung der FWA muss die benötigte Modbus-Registerliste in dem Konfigurationstool angegeben werden. Für die FWA muss zusätzlich die Globalstrahlung durch ein Pyranometer gemessen werden. Pro FWA wird nur ein Pyranometer benötigt, unabhängig von der Anzahl der Erzeugungsanlagen, aus diesem Grund wird von dem PV-Controller nur eins unterstützt. Als Pyranometer wird das sun[e] 00.16130.501030 des Herstellers Lambrecht meteo verwendet. Die unterstützten Pyranometer können analog zu den Wechselrichtern und Stromzählern, durch eine Erweiterung des Konfigurationstool um weitere Modbus-Registerlisten, ergänzt werden.

Bei einer zukünftigen Erhöhung der maximalen Anzahl der angeschlossenen Wechselrichter und Stromzähler ist darauf zu achten, dass die Performance des PV-Controllers nicht übermäßig beeinträchtigt wird. Ebenfalls sollten neben technischen Gründen auch kaufmännische Gründe bei der Erhöhung der maximalen Anzahl der angeschlossenen Wechselrichter und Stromzähler berücksichtigt werden.

4.3 Erstellen des Githubs

Zur Sicherung des Fortschrittes während der Programmierung und für die einfache Distribution der fertigen Software wird ein Software-Repository auf Basis des Open-Source-Projekts Git bei Github.com aufgesetzt. Dafür wird auf Github.com ein neues Repository erstellt, in dem das Projekt gespeichert wird [50]. Um mit Node-RED auf das Repository zuzugreifen, müssen Projekte in Node-RED aktiviert sein und ein neues SSH-Schlüsselpaar generiert werden. Dem SSH-Schlüsselpaar kann zur Erhöhung der Sicherheit eine Passphrase hinzugefügt werden. Diese muss beim Schreiben auf das Repository angegeben werden. Um nachzuvollziehen zu können, welche Änderungen von wem vorgenommen wurden, müssen der Benutzername sowie die E-Mail-Adresse angegeben werden. Der Dialog zum Erzeugen eines neuen SSH-Schlüsselpaars ist in der Abbildung 11

abgebildet. Der Public-Key des Schlüsselpaars wird anschließend in den Einstellungen des Repository hinterlegt und es wird Schreibzugriff auf das Repository gewährt. Nach erfolgreicher Einrichtung des Repositories ist der Programmierfortschritt gegen Datenverluste gesichert und kann aufwandsoptimiert auf eine Vielzahl von PV-Controllern installiert werden.

Einstellungen

Schließen

Ansicht

Git-Benutzerkonto

Leer lassen für Systemstandard

Palette

Benutzername

E-Mail

Git-Konfig.

Tastatur

Environment

Arbeitsablauf

Wählen Sie Ihren bevorzugten Git-Arbeitsablauf (Workflow)

☐ Manuell
Alle Änderungen müssen manuell übertragen werden (commit) über die Seitenleiste 'Projekthistorie'

☒ Automatisch
Änderungen werden automatisch übertragen (commit) mit jeder Übernahme (deploy)

SSH-Schlüssel

Ermöglicht es Ihnen, eine sichere Verbindungen zum (remote) Server-Repository herzustellen [Schlüssel hinzufügen](#)

SSH-Schlüssel hinzufügen

Ein neues öffentlich-privates Schlüsselpaar generieren

Name Darf nur A-Z 0-9 _ enthalten

Passphrase Optional

Abbildung 11: Konfiguration eines GitHub Repository in Node-RED

5 Programmierung des PV-Controllers

In den nachfolgenden Abschnitten wird das Vorgehen bei der Programmierung des PV-Controllers sowie dessen Funktionen beschrieben. Voraussetzung für die Programmierung ist, dass der RevPi, der auf dem der PV-Controller implementiert wird, vollständig geupdatet ist und Node-RED installiert wurde. Des Weiteren muss Node-RED zum Autostart des PV-Controllers hinzugefügt werden. Damit wird sichergestellt, dass der PV-Controller nach einer Spannungsunterbrechung selbständig wieder hochfährt und Node-RED weiter ausführt. Über den Konsolenbefehl `sudo systemctl enable nodered.service` wird Node-RED dem Autostart des RevPis hinzugefügt.

Der vollständige Quellcode des PV-Controller ist in dem Anhang D zu finden.

5.1 Konfiguration des PV-Controllers

Für die Konfiguration des PV-Controllers wird die csv-Konfigurationsdatei, die mittels des in Abschnitt 6 beschriebenen Excel-Konfigurationstools erzeugt wird, benötigt. Eine vollständige Konfigurationsdatei ist im Anhang E zu finden. Die erzeugte Konfigurationsdatei muss auf den PV-Controller übertragen werden. Beim Programmstart wird die Konfigurationsdatei einmalig vom PV-Controller eingelesen und die Konfiguration auf den PV-Controller übertragen. Denkbar wäre auch, dass die Konfigurationsdatei während der Laufzeit des Programms mehrfach eingelesen wird, wenn Daten aus dieser benötigt werden. Es wurde sich jedoch dagegen entschieden, da das Lesen von Dateien viel Rechenleistung benötigt und bei jeder Geräteabfrage über Modbus die Datei erneut gelesen werden muss. Das Einlesen der Konfigurationsdatei wird in dem `Config Flow` vorgenommen. Dieser ist in der Abbildung 12 dargestellt.

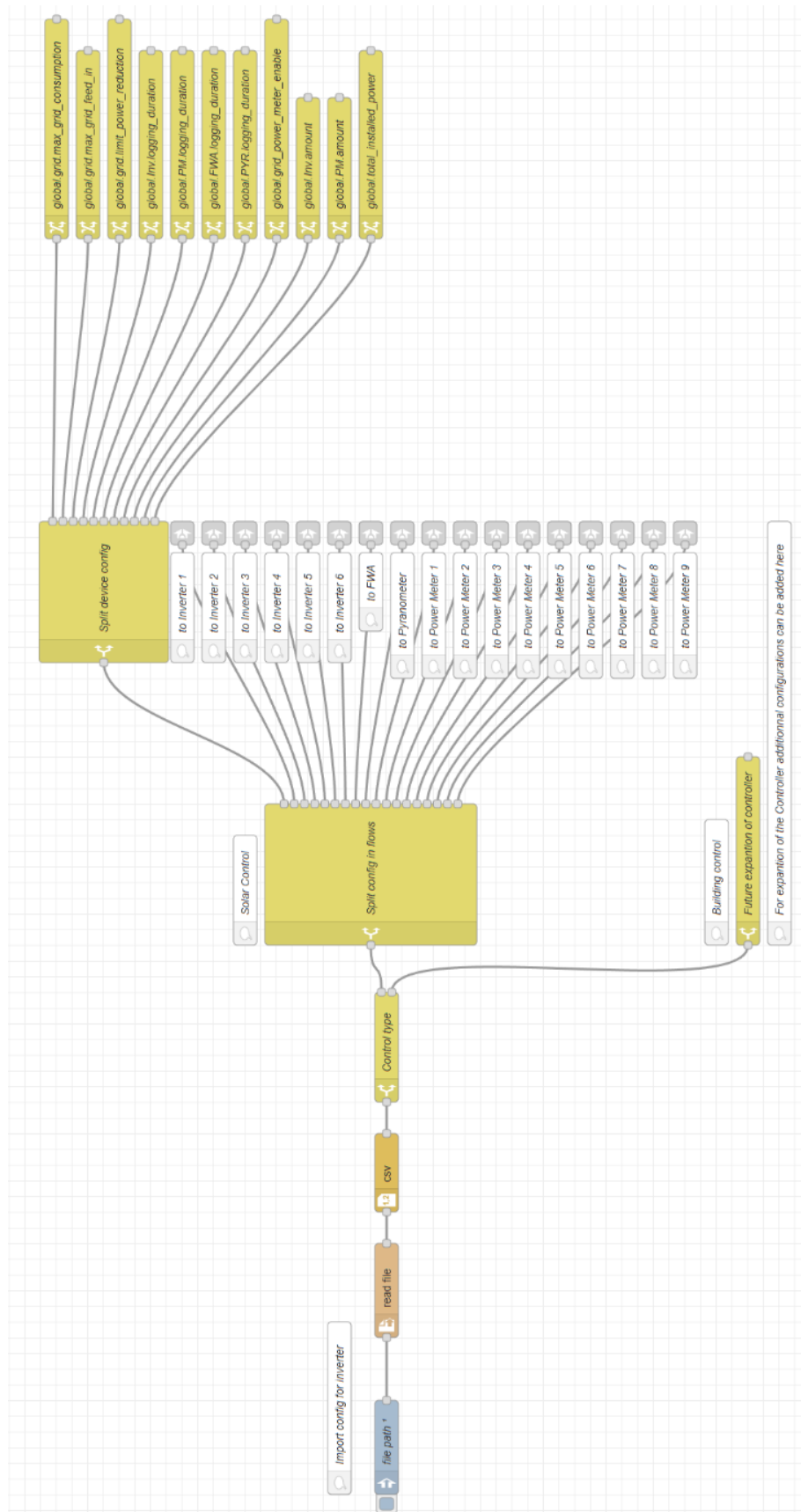


Abbildung 12: Node-RED Config Flow

Bei Programmstart wird automatisch von dem `Inject Node file path` der Dateipfad der Konfiguration an den `read file Node` gesendet. Dieser liest die Datei ein und sendet diese als UTF-8 Zeichenfolge an den `CSV-Parser Node` weiter. Der `CSV-Parser` gibt die csv-Konfigurationsdatei zeilenweise als `msg.object` aus. Eine Zeile der Konfigurationsdatei enthält vier Spalten, über die eine Zuordnung an die richtige Variable vorgenommen wird. In der Tabelle 5 sind zwei Beispielzeilen der Konfigurationsdatei abgebildet, an denen sich die Funktionsweise des Einlesens der Konfiguration erklären lässt.

Tabelle 5: Zwei Beispiel Zeilen der Konfigurationsdatei

Spalte 1 Teilbereich	Spalte 2 Flow	Spalte 3 Bezeichner	Spalte 4 Wert
Solar control	Config	Max grid consumption	270
Solar control	Inv 1	Inv Yield add	30775

Die Zuordnung der Konfiguration an den richtigen Speicherort geschieht über die Eintragungen in den ersten drei Spalten der Konfigurationsdatei. In der ersten Spalte wird der Teilbereich des PV-Controllers angegeben, in dieser Arbeit sind alle Konfigurationen dem Teilbereich „Solar control“ zuzuordnen. Die zweite Spalte gibt den Flow an, an welchen das `msg.object` weitergeleitet werden muss, und die dritte Spalte gibt den Bezeichner der Variable an.

Es werden die Eintragungen der Spalten nacheinander von `Switch Nodes` ausgewertet. In den `Switch Nodes` werden die `msg.objects` abhängig vom Inhalt der Spalten an einen Ausgang des Nodes weitergeleitet. In der Abbildung 13 ist die Konfiguration des `Switch Nodes Split config in flows` zur Weiterleitung der Konfiguration an den richtigen Flow dargestellt, abhängig von dem Inhalt der zweiten Spalte.

Node 'switch' bearbeiten

Löschen Abbrechen Fertig

Eigenschaften

Name Split config in flows

Eigenschaft msg.payload.col2

==	▼ a _z Config	→ 1 ✕
==	▼ a _z Inv 1	→ 2 ✕
==	▼ a _z Inv 2	→ 3 ✕
==	▼ a _z Inv 3	→ 4 ✕
==	▼ a _z Inv 4	→ 5 ✕
==	▼ a _z Inv 5	→ 6 ✕
==	▼ a _z Inv 6	→ 7 ✕
==	▼ a _z Cp 1	→ 8 ✕
==	▼ a _z Cp 2	→ 9 ✕
==	▼ a _z Cp 3	→ 10 ✕
==	▼ a _z Cp 4	→ 11 ✕
==	▼ a _z Cp 5	→ 12 ✕
==	▼ a _z Cp 6	→ 13 ✕

Abbildung 13: Switch Node Split config in flows aus Abbildung 12

So würden die erste in der Tabelle 5 gezeigte Beispielzeile der Konfigurationsdatei an den Ausgang 1 des Nodes weitergeleitet werden und die zweite Zeile wird an den Ausgang 2 des Nodes weitergeleitet.

Alle Konfigurationen, die an den Ausgang 1 des Switch Nodes Split config in flows weitergeleitet werden, werden an mehreren Stellen im Programm benötigt. Nach der Weiterleitung durch den Switch Nodes Split device config werden diese mit einem Change Node als globale Variable gespeichert. Dafür wird, wie in Abbildung 14 dargestellt, der Inhalt der vierten Spalte (Wert) als globale Variable übernommen. Somit kann aus jedem Flow auf diese zugegriffen werden.

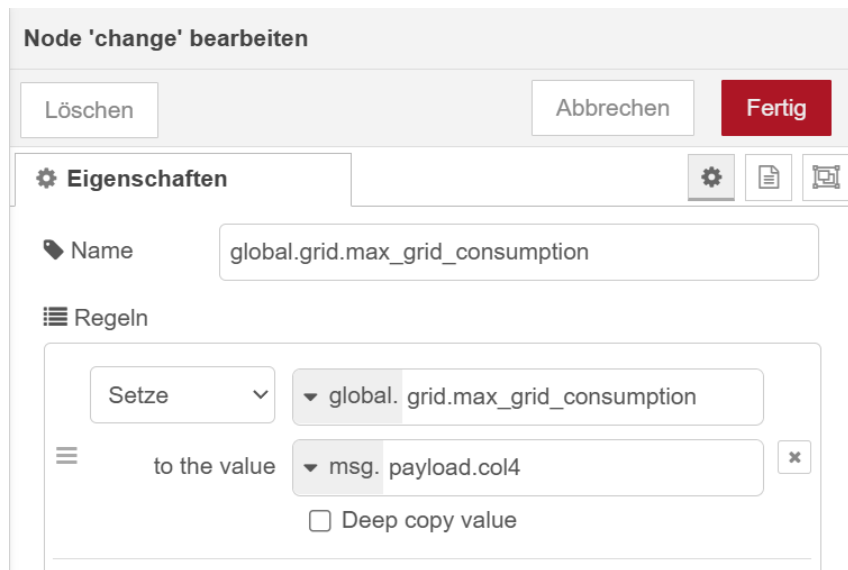


Abbildung 14: Einstellungen eines Change Nodes

Bei einer Weiterleitung des `msg.objects` durch den in Abbildung 12 dargestellten Switch Node Split config in flows an einen anderen Ausgang als den ersten, werden die Daten an den jeweiligen Flow mit Hilfe von Link Nodes übertragen. Diese Konfigurationen werden nur in dem jeweiligen Flow benötigt und werden deshalb als Flow Variable abgespeichert. Auf Flow-Variablen kann nur im jeweiligen Flow zugegriffen werden. So wird die zweite Zeile der in Tabelle 5 dargestellten Zeile der Konfigurationsdatei an den Inverter 1 Flow weitergeleitet. Nach der Zuordnung auf Grund des Eintrages in der dritten Spalte durch einen weiteren Switch Node wird die Konfiguration in dem Flow durch Change Nodes abgespeichert. In der Abbildung 15 ist das Speichern der Konfiguration im Inverter 1 Flow dargestellt.

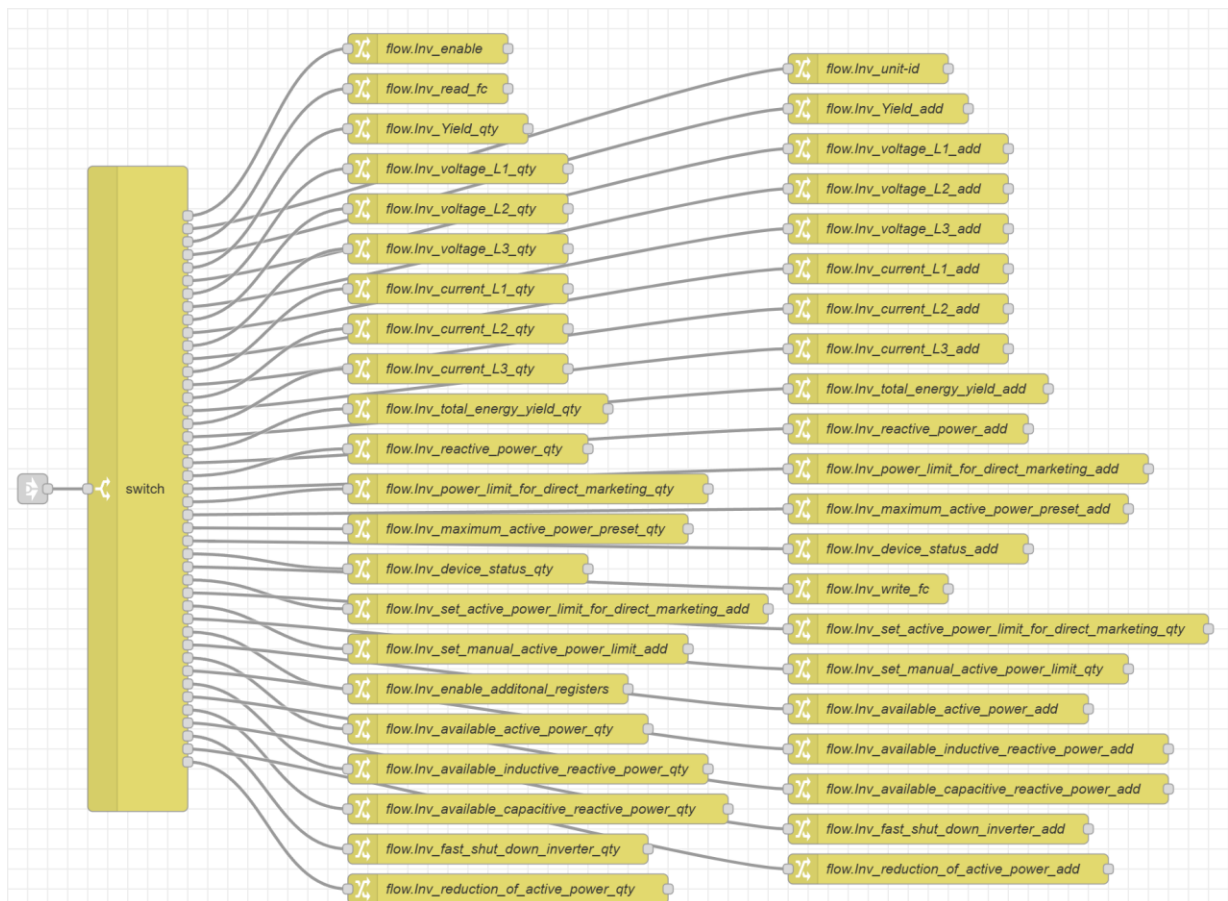


Abbildung 15: Ausschnitt Inverter 1 Flow: Speichern der Konfiguration

Das Einlesen der restlichen Konfiguration geschieht nach demselben Prinzip. Parallel zum Einlesen der Konfigurationsdatei werden auf dem PV-Controller Merker-Variablen erzeugt. Diese werden vom PV-Controller benötigt, um sich Zustände während des Programmablaufes zu merken. Dadurch, dass sich die Merker-Variablen nur Zustände merken oder Messwerte zwischenspeichern, müssen diese nicht durch die Konfiguration auf den PV-Controller geladen werden. Die Merker-Variablen werden in den nachfolgenden Abschnitten erläutert, sobald sie für eine Funktion des PV-Controllers verwendet werden.

Neben der selbständig eingelesenen Konfiguration muss für die korrekte Funktion des PV-Controllers, die in Abschnitt 6.9 Konfiguration übertragen beschriebene, händische Konfiguration durchgeführt sein.

5.2 Main Flow

Nachdem die Konfiguration des PV-Controllers abgeschlossen ist, beginnt der PV-Controller zyklisch zu arbeiten. Dieser zyklische Ablauf wird von dem `Main Flow`, siehe Abbildung 16, koordiniert.

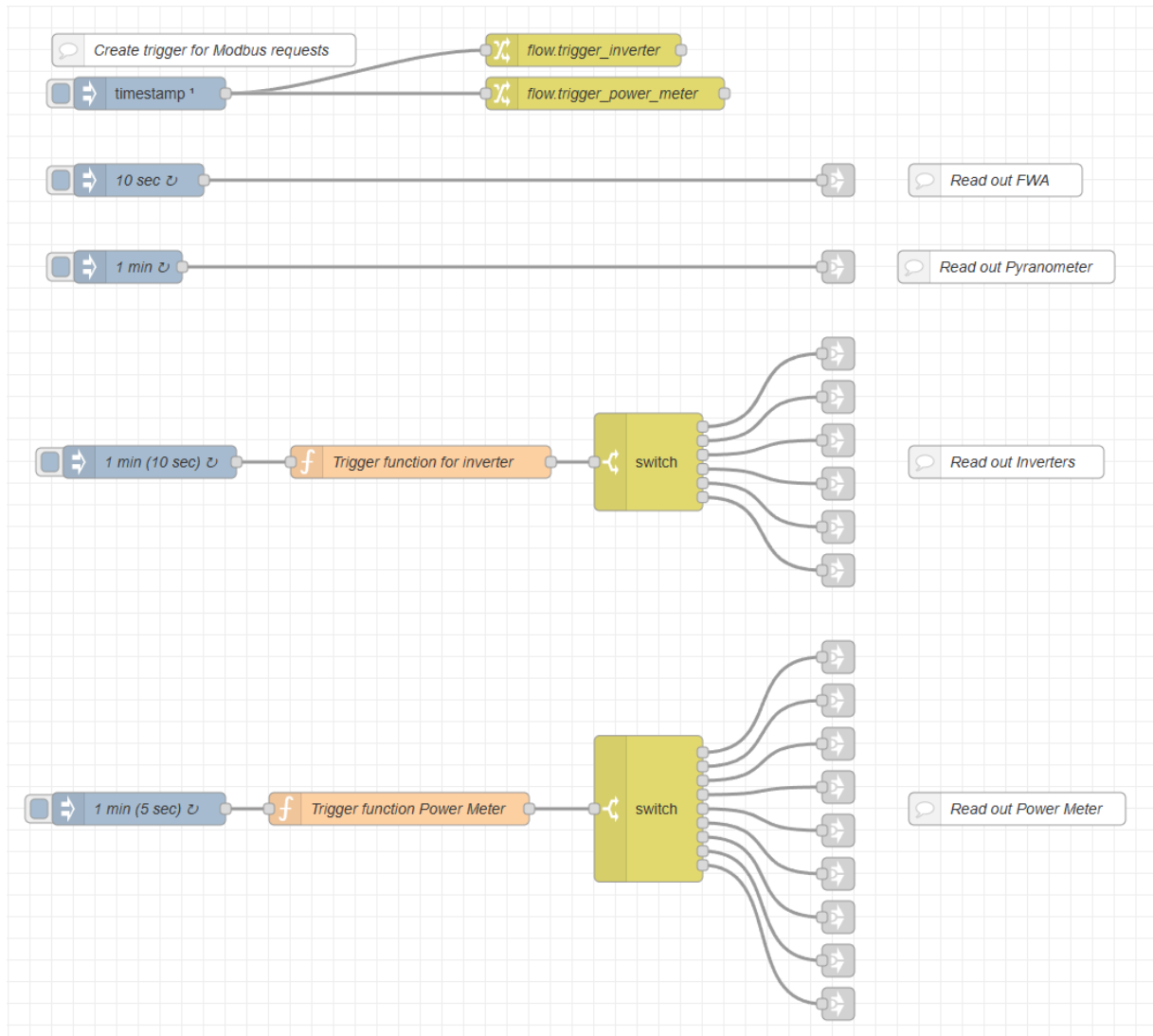


Abbildung 16: Main Flow

Im `Main Flow` sind die angeschlossenen Geräte abhängig ihres Typs in Gruppen unterteilt. Diese Gruppen werden zyklisch abgefragt. Dabei gibt es die Gruppen FWA, Pyranometer, Wechselrichter und Stromzähler. Die Abfrage wird jeweils durch ein `Inject Node` gestartet.

Die Gruppen FWA und Pyranometer bestehen dabei aus jeweils nur einem Gerät. Die Abfrage der FWA wird durch den `Inject Node` alle 10 Sekunden gestartet. Dieses Abfrageintervall wurde gewählt, um ausreichend Zeit für die Umsetzung der Steuersignale zu haben. Diese müssen nach spätestens 60 Sekunden umgesetzt sein. Die Abfrage des Pyranometers ist nicht zeitkritisch und es wurde ein Abfrageintervall von einer Minute gewählt. Wenn keine FWA oder kein Pyranometer angeschlossen ist, wird die Abfrage, nach der Weiterleitung durch die `Link out Nodes` in den jeweiligen Flow, durch einen `Switch Node` gestoppt. Diese leiten das `msg.object` nur weiter, wenn die in der Konfiguration übergebenen Flow Variable `FWA_enable` bzw. `Pyr_enable` den Wert 1 haben. In der Abbildung 17 ist der `Switch Node` der FWA dargestellt.

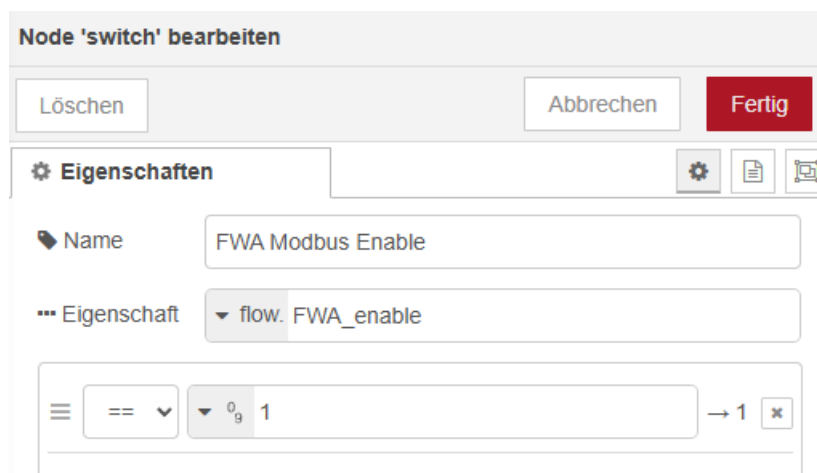


Abbildung 17: Switch Node FWA Modbus Enable aus dem FWA Flow

Die Gruppen Wechselrichter und Stromzähler bestehen aus mehr als einem Gerät. Um diese hintereinander abzufragen, wird während der Konfiguration jeweils eine Merker-Variable erzeugt, mit der sich das zuletzt abgefragte Gerät gemerkt wird. Die Merker-Variablen werden im ganz oben im `Main Flows` erzeugt und mit dem Wert 0 als Flow-Variable gespeichert. Der Merker für die Wechselrichter heißt `flow.trigger_inverter` und der Merker für die Stromzähler heißt `flow.trigger_power_meter`. Die zyklische Abfrage funktioniert für die Gruppe Wechselrichter und Stromzähler nach dem gleichen Prinzip. Ein `Inject Node` gibt zyklisch eine `msg.object` an den nachfolgenden `Function Node` weiter. In der Abbildung 18 ist der `Function Node` der Gruppe der

Wechselrichter dargestellt. Der Function Nodes der Gruppe der Stromzähler unterscheidet sich dadurch, dass der Merker `flow.trigger_power_meter` anstelle des `flow.trigger_inverter` Merker verwendet wird und die Anzahl der Stromzähler `global.PM_amount` anstelle der Anzahl der Wechselrichter `global.Inv-amount` in dem Function Node verwendet wird.

Die Aufgabe dieser Function Nodes ist es, die Abfrage von nicht konfigurierten Geräten zu verhindern. Der PV-Controller wird dadurch skalierbar und die Anzahl der angeschlossenen Geräte kann in der Konfiguration verändert werden.

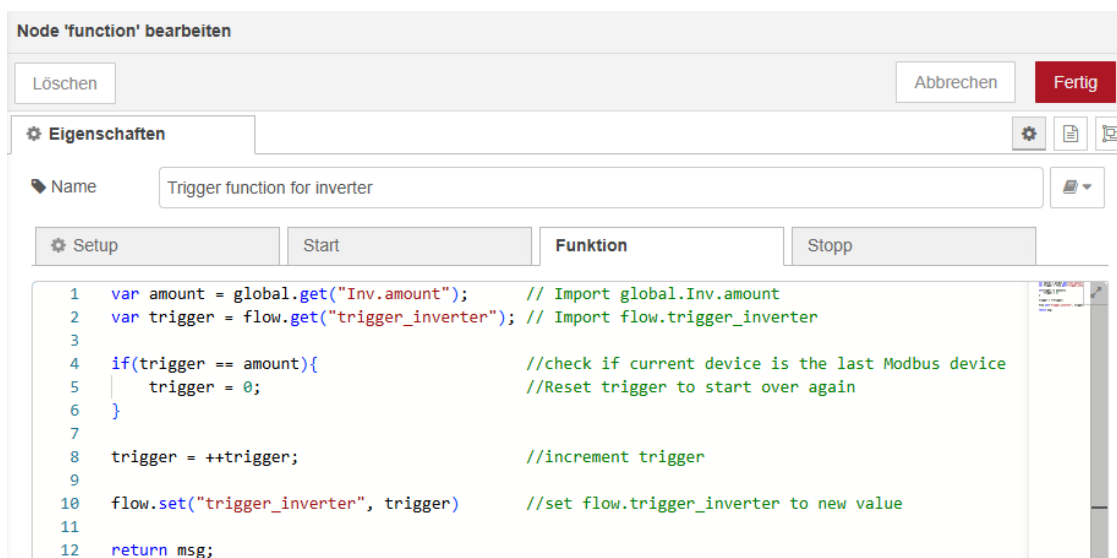


Abbildung 18: Function Node Trigger function for Inverter der zyklischen Abfrage der Wechselrichter

In diesem Function Node wird der Merker `flow.trigger_inverter` zu Beginn mit der Anzahl der angeschlossenen Wechselrichter `global.Inv_amount` verglichen. Falls diese übereinstimmen, wird der Merker `flow.trigger_inverter` auf 0 zurückgesetzt und die Abfrage beginnt erneut beim ersten Wechselrichter der Gruppe. Falls diese nicht übereinstimmen, wird der Merker `flow.trigger_inverter` nicht zurückgesetzt und die Abfrage der Gruppe wird fortgesetzt. Nach dem Vergleichen wird der Merker `flow.trigger_inverter` um 1 erhöht.

Anschließend wird das `msg.object` an den nachfolgenden Switch Node weitergeleitet. In dem Switch Node wird das `msg.object` abhängig von dem Zustand des zugehörigen

Merker an den jeweiligen Ausgang weiter geleitet. Die Konfiguration des `Switch Node` für die Abfrage der Wechselrichter ist in der Abbildung 19 dargestellt.

Node 'switch' bearbeiten

Löschen Abbrechen Fertig

Eigenschaften

Name

Eigenschaft flow. trigger_inverter

≡	=	▼	a _z 1	→ 1	x
≡	=	▼	a _z 2	→ 2	x
≡	=	▼	a _z 3	→ 3	x
≡	=	▼	a _z 4	→ 4	x
≡	=	▼	a _z 5	→ 5	x
≡	=	▼	a _z 6	→ 6	x

Abbildung 19: Switch Node für die Abfrage der Wechselrichter

Durch `Link out Nodes` werden die `msg.objects` an die jeweiligen Flows zum Starten der Modbus-Datenabfragen übertragen.

Die Intervalle, mit denen die Abfrage der Gruppen der Wechselrichter und der Stromzähler gestartet wird, wurden so gewählt, dass diese mindestens einmal in der Minute abgefragt werden. Beim Anschluss von weniger Geräten als maximal möglich, verringert sich das Abfrageintervall. Durch ein verkürztes Abfrageintervall erhöht sich jedoch nicht die Prozessorlast, da die Geräte sequenziell abgefragt werden.

5.3 Modbus-Kommunikation

In den folgenden Abschnitten wird die Modbus-Kommunikation beschrieben. Zunächst wird auf die Datenabfrage über Modbus eingegangen. Anschließend wird das Senden von Daten über Modbus beschrieben.

5.3.1 Modbus-Datenabfrage

Zunächst wird die Realisierung der Modbus-Datenabfrage mit den an den PV-Controller angeschlossenen Geräten, anhand des in Abbildung 20 dargestellten `Inverter 1 Flows`, beschrieben. Die Abfrage der anderen angeschlossenen Geräte folgt dem gleichen Aufbau wie der `Inverter 1 Flow`. Es werden lediglich andere Datenpunkte abgefragt. Der Flow wird aus dem `Main Flow` gestartet über den `Link In Node`, dieser befindet sich in der oberen linken Ecke des Flows.

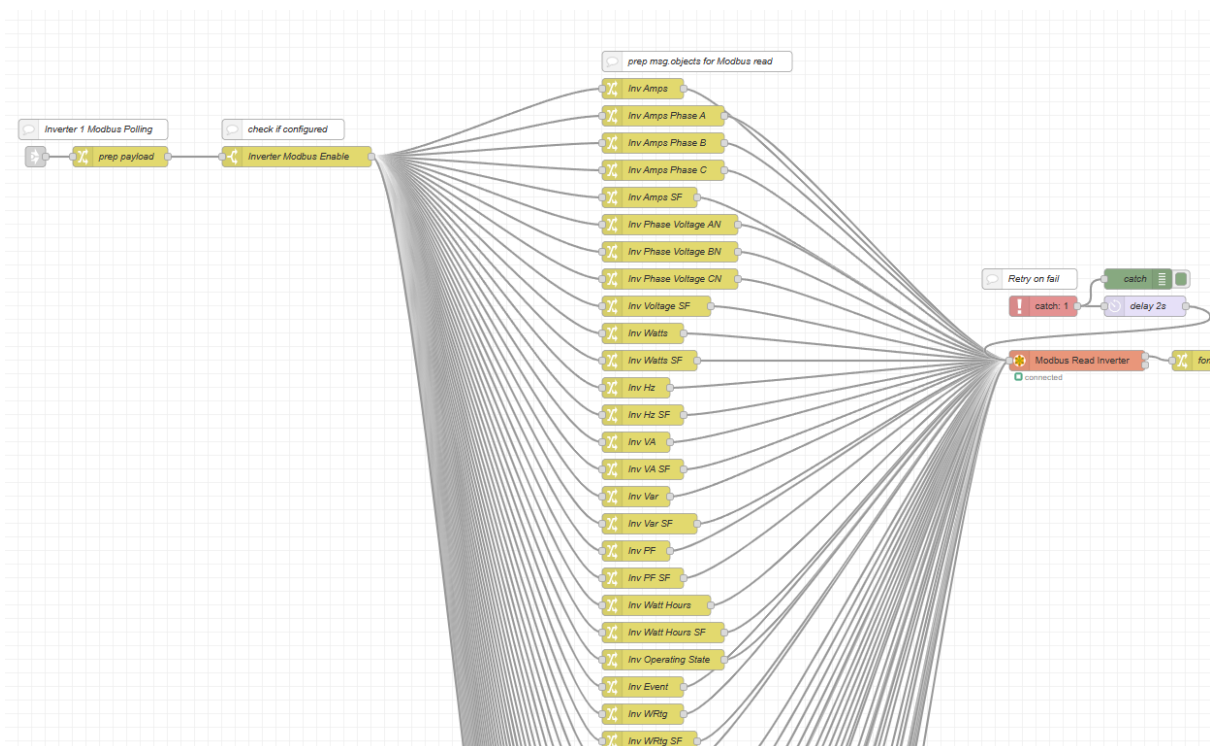


Abbildung 20: Modbus-Kommunikation Inverter 1 Flow Teil 1

Das aus dem `Main Flow` kommende `msg.object` wird zunächst in dem `Change Node` `prep payload` formatiert. Dies wird getan, um die Struktur des `msg.objects` für die

später folgende Modbus-Datenabfrage vorzubereiten. Nach dem `Change Node prep payload` wird das `msg.object` an den `Switch Node Inverter Modbus Enable` weitergeleitet. Dieser Node überprüft, ob für das Gerät eine gültige Konfiguration vorliegt. Ist das nicht der Fall, so wird der Flow an dieser Stelle stoppen. Wenn keine gültige Konfiguration vorliegt, ist die Variable `flow.Inv_enable` auf *false* gesetzt. Anschließend wird das `msg.object` gleichzeitig an mehrere `Change Nodes` übertragen. Es existiert für jedes abzufragendes Register ein `Change Node`. Zur Übersichtlichkeit sind in der Abbildung 20 nicht alle dieser `Change Nodes` dargestellt, die nicht abgebildeten `Change Nodes` übernehmen die gleiche Aufgabe wie die anderen `Change Nodes`. In den `Change Nodes` wird das vorbereitete `msg.object` mit den für die Modbus-Abfrage benötigten Daten befüllt. Die benötigten Daten wurden bereits in dem Abschnitt 3.4 Benötigte Daten für die Modbus-Kommunikation herausgearbeitet. In der Abbildung 21 ist dies anhand des `Change Node Inv Amps` dargestellt.

The screenshot shows the configuration for a 'change' node named 'Inv Amps'. The interface includes buttons for 'Löschen', 'Abbrechen', and 'Fertig'. Below the 'Eigenschaften' tab, the 'Regeln' section contains five rules, each mapping a message field to a flow variable:

- Rule 1: Setze `msg.topic` to the value `flow.Inv Amps`.
- Rule 2: Setze `msg.payload.fc` to the value `flow.Inv_read_fc`. The 'Deep copy value' checkbox is unchecked.
- Rule 3: Setze `msg.payload.address` to the value `flow.Inv_Amps_add`. The 'Deep copy value' checkbox is unchecked.
- Rule 4: Setze `msg.payload.quantity` to the value `flow.Inv_Amps_qty`. The 'Deep copy value' checkbox is unchecked.
- Rule 5: Setze `msg.payload.unitid` to the value `flow.Inv_unit_id`. The 'Deep copy value' checkbox is checked.

Abbildung 21: Change Node Inv Amps

Nachdem die `msg.objects` durch die `Change Nodes` angepasst wurden, werden alle `msg.objects` an den `Modbus-Flex-Getter Node Inverter Modbus` weitergeleitet. Von dem `Modbus-Flex-Getter Node Inverter Modbus` wird die Modbus-Abfrage des Wechselrichters übernommen und koordiniert, damit nicht zu viele Modbus-Abfragen zeitgleich gestellt werden und es zu Buskollisionen kommt. In dem `Modbus-Flex-Getter Node Inverter Modbus` ist die für die Abfrage benötigte IP-Adresse und Port des abzufragenden Wechselrichters hinterlegt. Diese sind für jedes Gerät individuell und werden während der Konfiguration des PV-Controllers für jedes angeschlossene Gerät angepasst. Die abgefragten Register werden nach der erfolgreichen Abfrage einzeln von dem `Modbus Flex Getter Node Modbus Inverter` als einzelne `msg.object` ausgegeben. In dem Fall, dass die Modbus-Abfrage fehlschlägt, wird von dem `catch Node` die Abfrage erneut gestartet, nach einer Verzögerung von 2 Sekunden durch einen `delay Node`. Gleichzeitig wird über einen `debug Node` eine Meldung über die fehlgeschlagene Modbus-Datenabfrage ausgegeben.

Der folgende Teil des Flows ist in der Abbildung 22 dargestellt. Nachdem von dem `Modbus Flex Getter Node Modbus Inverter` die abgefragten Register ausgegeben werden, wird in dem nachfolgenden `Change Node` das `msg.object` erneut formatiert, damit die Daten der Register in der richtigen Form für die Weiterverarbeitung und das Logging vorliegen

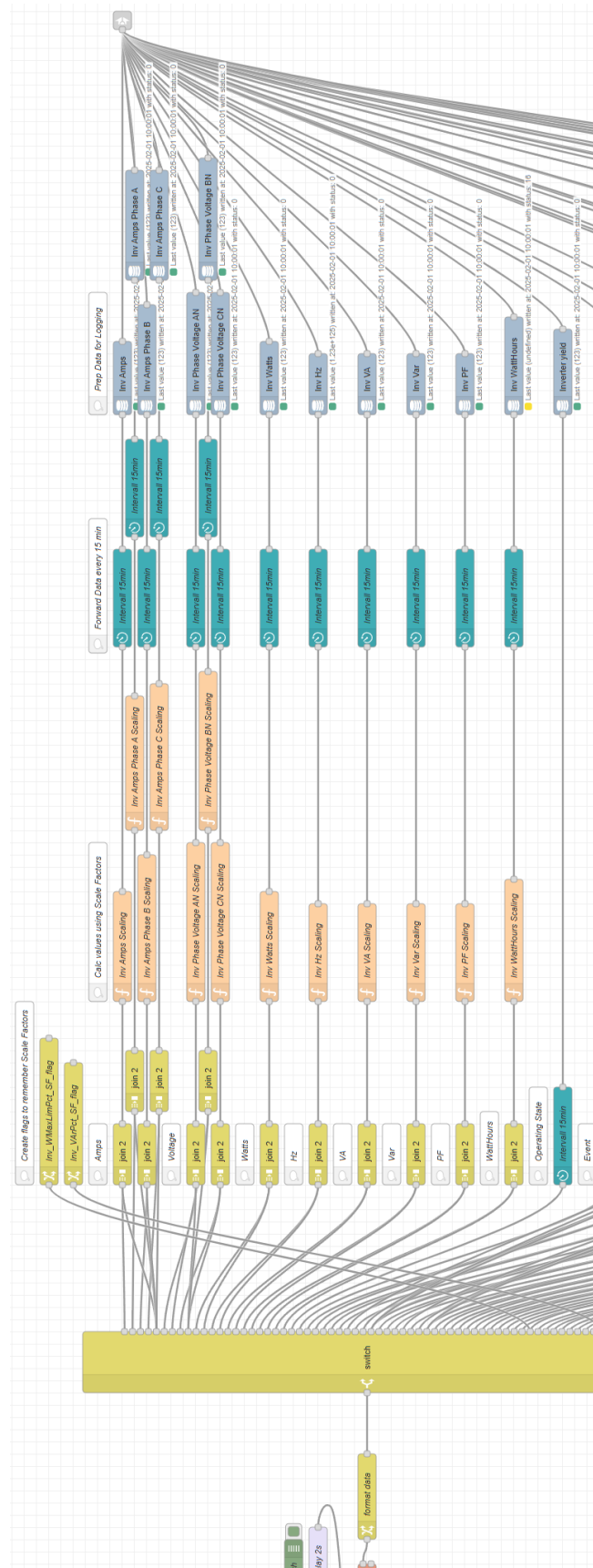


Abbildung 22: Modbus-Kommunikation Inverter 1 Flow Teil 2

Anschließend werden auf Grundlage des `msg.topics` der `msg.objects` diese von einem Switch Node an den richtigen Ausgang weitergeleitet. An der folgenden Stelle im Flow unterscheidet sich die Modbus-Abfrage der Wechselrichter von der Modbus-Abfrage der Stromzähler und der FWA. Bei den Wechselrichtern müssen die Daten aus den abgefragten Registern noch mit den ebenfalls abgefragten Skalierungsfaktoren skaliert werden.

Das Vorgehen zur Skalierung der Daten wird im Folgenden an dem Datenpunkt `Inv Amps` beschrieben. Wie in Abbildung 20 zu erkennen ist, wurden zunächst die Register mit dem Wert von `Inv Amps` und der Skalierungsfaktor `Inv Amps SF` über Modbus abgefragt. Nach dem Switch Node aus Abbildung 22 werden die beiden `msg.Objects` von `Inv Amps` und `Inv Amps SF` mit einem Join Node zu einem `msg.Object` zusammengefasst. Dies wird getan, um sicherzustellen, dass beide Register bereits abgefragt wurden. Anschließend wird das `msg.object` an den Function Node `Inv Amps Scaling` weitergeleitet.

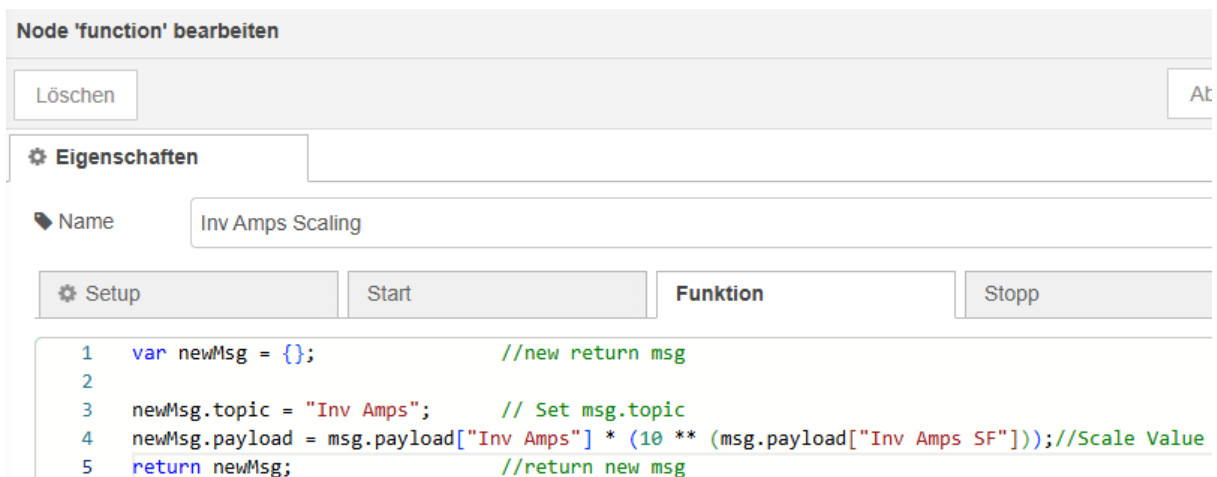


Abbildung 23: Function Node `Inv Amps Scaling`

In dem in Abbildung 23 dargestellten Function Node `Inv Amps Scaling` wird ein neues `msg.object` erstellt. Diesem wird ein `msg.topic` sowie als `msg.payload` der skalierte Wert von `Inv Amps` hinzugefügt. Zuletzt wird das neu erstellte `msg.object` zurückgegeben und zum `Intervall Node` weitergeleitet. Die Skalierungsfaktoren `Inv_WMaxLimPct_SF` und `Inv_VArPct_SF` werden für das Senden der Steuerbefehle an die FWA benötigt, deshalb werden diese in den Merker-Variablen `Inv_WMaxLimPct_SF_flag` und `Inv_VArPct_SF_flag` zwischengespeichert. Die abgefragten Daten des Pyranometers

müssen ebenfalls skaliert werden, jedoch sind die Skalierungsfaktoren fest und werden in der Konfiguration mit übergeben.

In der Abbildung 24 ist ein Teil des Power Meter 1-3 Flow dargestellt. Dieser und auch der FWA Flow kommen ohne eine Skalierung aus. Von dem Switch Node werden die `msg.objects` direkt an die Intervall Nodes weitergeleitet.

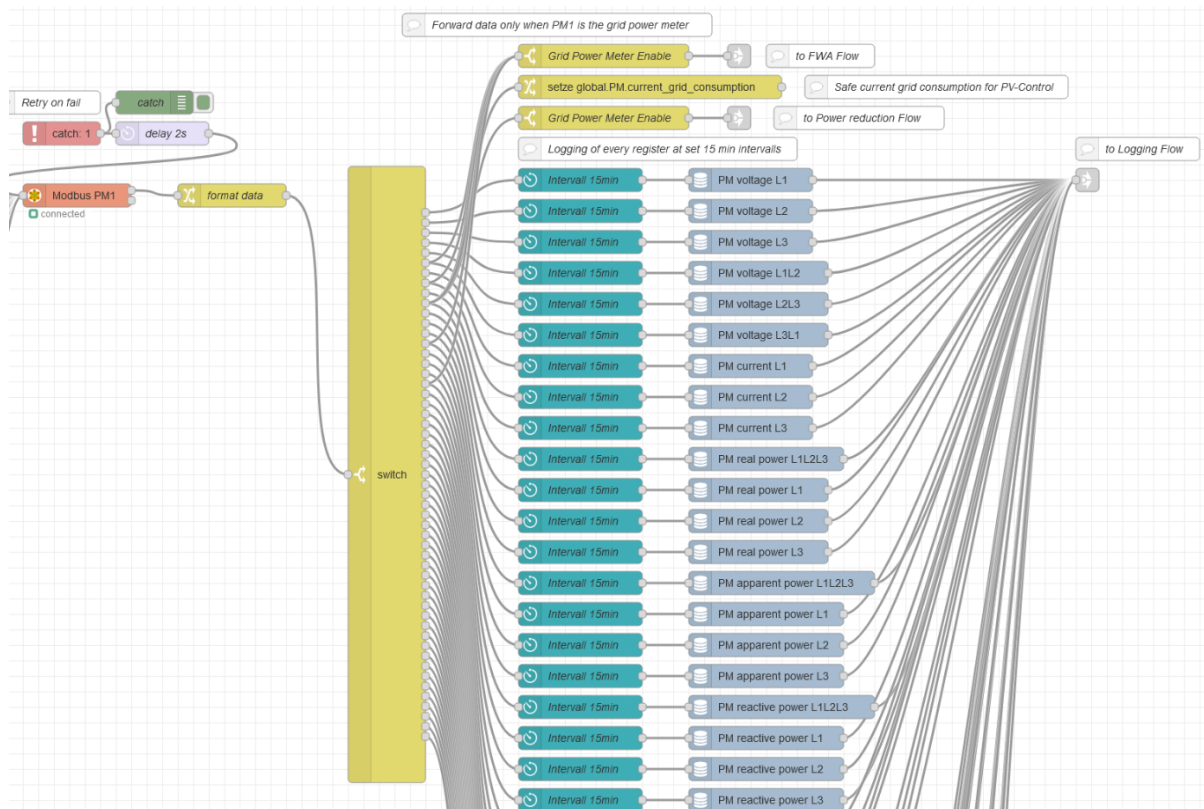


Abbildung 24: Modbus-Kommunikation ohne Skalierung im Power Meter 1-3 Flow

Bei beiden Varianten werden Daten direkt bevor sie auf den Intervall Node treffen mit Hilfe eines Link out Nodes abgegriffen. Diese Daten werden für unterschiedliche Aufgaben benötigt, in der Abbildung 24 sieht man, dass dem FWA Flow alle benötigten Daten über den NAP übergeben werden. Ab dem Intervall Node beginnt das Logging für JEVIs, dies wird in dem Abschnitt 5.4 Logging der Daten für JEVIs weiter beschrieben.

5.3.2 Modbus Daten senden

Neben der Datenabfrage von den angeschlossenen Geräten müssen auch Daten über Modbus gesendet werden. Dabei müssen die Daten an die FWA und die Wechselrichter übermittelt werden. Die Stromzähler und auch das Pyranometer müssen keinerlei Daten übergeben bekommen, sie werden lediglich abgefragt.

An die FWA werde folgende Daten übertragen: die Rückmeldung über die Umsetzung der Steuersignale, die Globalstrahlung gemessen vom Pyranometer und die Messwerte des Bezugszählers. Dies ist in der Abbildung 25 dargestellt.

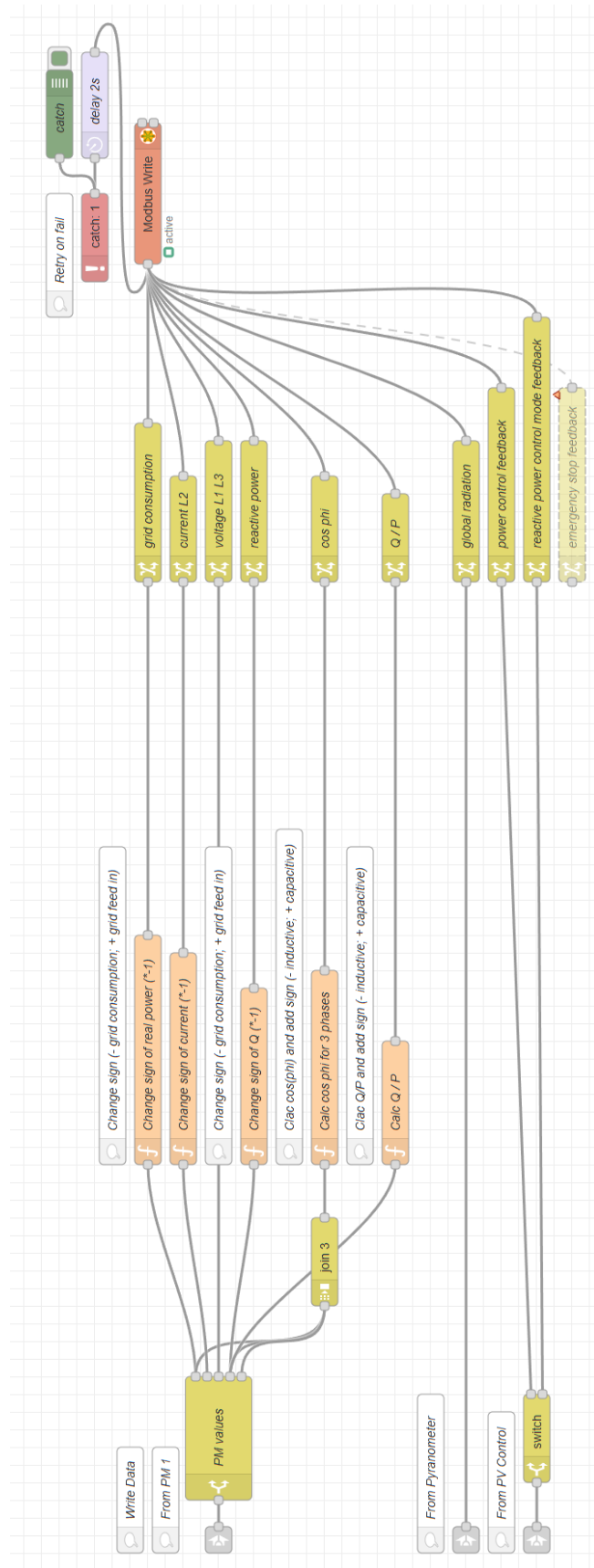


Abbildung 25: FWA Flow Modbus Daten senden

Über jeweils einen `Link in Node` werden die für die FWA benötigten Datenpunkte in den Flow übertragen. Wenn aus einem Flow mehrere Datenpunkte übertragen werden, müssen diese anschließend durch einen `Switch Node`, anhand ihres `msg.topic`, sortiert werden. In dem `FWA Flow` ist dies für die Datenpunkte aus dem `Power Meter 1-3 Flow` und aus dem `PV Control Flow` der Fall. Aus den Datenpunkten des `Power Meter 1-3 Flows` werden noch für die FWA, der Leistungsfaktor $\cos(\phi)$ und Q/P_{inst} in % berechnet und deren Vorzeichen an die Anforderungen der WEMAG aus Tabelle 2 angepasst. Durch die `Join Nodes` wird sichergestellt, dass alle für die Berechnung von der Phasenverschiebung $\cos(\phi)$ und Q/P_{inst} in % benötigten Daten vorliegen. Die Vorzeichen von der Wirkleistung, der Blindleistung und des Stroms müssen ebenfalls angepasst werden. Dies wird von den `Function Nodes` übernommen.

Anschließend werden die `msg.objects` zu `Change Nodes` weitergeleitet. In den `Change Nodes` werden dem `msg.objects` die für die Modbus-Kommunikation benötigten Daten angehängt. In der Abbildung 26 ist dies für den Datenpunkt `global radiation` dargestellt.

Eigenschaften

Name

global radiation

Regeln

Setze

▼ msg. value

to the value

▼ msg. payload

☐ Deep copy value

Lösche

▼ msg. payload

Setze

▼ msg. payload

to the value

▼ {} {}

...

Setze

▼ msg. payload.value

to the value

▼ msg. value

☐ Deep copy value

Lösche

▼ msg. value

Setze

▼ msg. payload.fc

to the value

▼ flow. FWA_write_fc

☐ Deep copy value

Setze

▼ msg. payload.unitid

to the value

▼ flow. FWA_unit_id

☐ Deep copy value

Setze

▼ msg. payload.address

to the value

▼ flow. FWA_global_radiation_add

☐ Deep copy value

Setze

▼ msg. payload.quantity

to the value

▼ flow. FWA_global_radiation_qty

☐ Deep copy value

Abbildung 26: Change Node global radiation

Von den Change Nodes werden die `msg.objects` weiter an den Modbus Flex Write Node gesendet. Dieser arbeitet, wie auch der Modbus Flex Read Node, hintereinander die `msg.objects` am Eingang des Nodes ab und sendet die Daten an die FWA. Ein catch Node wiederholt im Fehlerfall ebenfalls die Übertragung und gibt eine Meldung über den debug Node aus.

Das Senden der Daten an die Wechselrichter ist vom Aufbau her sehr ähnlich zum FWA Flow. In der Abbildung 27 ist der Inverter 1 Flow dargestellt. Über einen Link in Node werden die vom PV Control Flow ausgegebenen Steuersignale in den Flow übertragen. Anschließend wird mit einem Switch Node überprüft, ob der Wechselrichter angeschlossen ist. Falls dies nicht der Fall ist, bricht das Senden ab. Grund dafür ist, dass die Steuerbefehle an alle der sechs Inverter Flows übergeben werden, unabhängig davon, ob diese angeschlossen sind. Anschließend werden die `msg.objects` anhand ihres `msg.topic` durch einen weiteren Switch Node weitergeleitet. In dem Inverter 1 Flow werden ebenfalls Function Nodes verwendet, diese werden genutzt, um die Werte `Inv_WMaxLimPct` und `Inv_VArWMaxPct` zu skalieren, mit den zwischengespeicherten Skalierungsfaktoren. Andernfalls würde dem Wechselrichter ein falsch formatierter Wert geliefert werden. Die restlichen Nodes übernehmen die gleichen Aufgaben wie die des FWA Flows aus Abbildung 25.

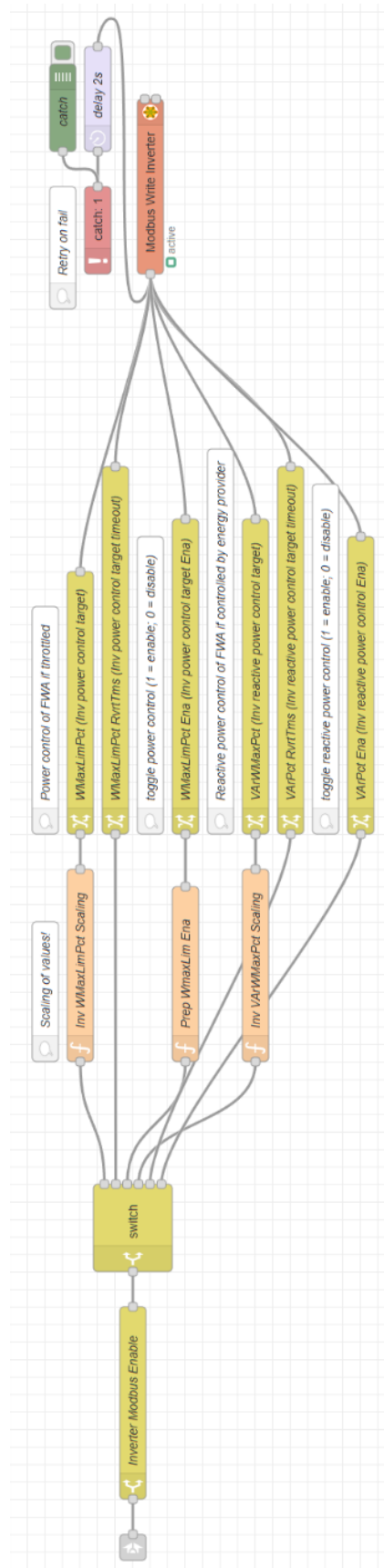


Abbildung 27: Inverter 1 Flow Modbus Daten senden

5.4 Logging der Daten für JEVIs

Nachdem die Daten über Modbus von den angeschlossenen Geräten abgefragt sind, müssen diese noch in den Ringspeicher des PV-Controllers geschrieben werden, sodass die Daten von JEVIs abgerufen werden können. Für das Logging der Daten werden, die von der Envidatec entwickelten Paletten zur Erfassung der Daten verwendet. Das Logging der Daten ist für jedes Gerät identisch und wird im Folgenden weiter anhand des `Inverter 1 Flow` beschreiben.

Das Logging der Daten für JEVIs beginnt ab den in Abbildung 22 dargestellten `Intervall` Nodes. Ein Teil des Bereiches des Loggings im `Inverter 1 Flow` ist in der Abbildung 28 vergrößert dargestellt.

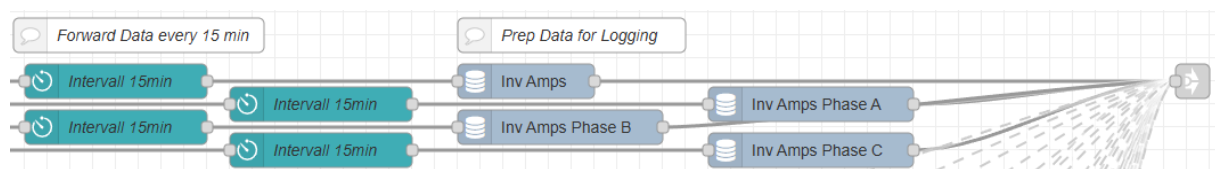


Abbildung 28: Inverter 1 Flow Logging

Die `msg.objects` mit dem abgefragten Register werden an die `Intervall` Nodes übertragen. Diese Nodes blockieren die `msg.objects` und lassen nur ein `msg.object` nach dem eingestellten Intervall passieren. Es wird immer das zuletzt an dem `Intervall` Node anliegende `msg.object` weitergereicht. Die anderen `msg.objects` werden nicht betrachtet. Die `Intervall` Nodes werden über einen Cron Befehl konfiguriert. Cron Befehle werden auf UNIX basierten Systemen verwendet, um wiederkehrende Aufgaben zu automatisieren. Entsprechend der Designvorgaben von Envidatec aus Abschnitt 2.2 werden die `Intervall` Node so konfiguriert, dass alle zu jeder Viertelstunde ein `msg.object` weiterreichen. Dafür wurde der Cron Befehl `*/15****` in dem Node hinterlegt. Das `msg.object` wird anschließend in dem `Logger` Node für die Speicherung im Ringspeicher des PV-Controllers vorbereitet. Dafür wird dem `msg.object` eine `Trend ID` zur Identifikation der Datenreihe angehängt sowie die Dauer des Verbleibs der Daten im Ringspeicher festgelegt. Die Dauer des Verbleibs der Daten im Ringspeicher kann in der

Konfiguration für den jeweiligen Gerätetyp festgelegt werden. Typisch ist eine Dauer von 60 Tagen. Es wird ebenfalls in dem Node die Datenbank ausgewählt, in der die Daten gespeichert werden soll. Diese Datenbank wird einmalig bei der Programmierung erstellt und in den `Logger` Nodes hinterlegt. Beim Erstellen der Datenbank muss lediglich ein Name vergeben werden, der innerhalb von Node-RED genutzt wird und ein Pfad mit Dateinamen, an dem die Datenbank auf dem PV-Controller gespeichert wird. Es wurde der Name `SQL-Lite DB` vergeben für die Datenbank innerhalb von Node-RED und es wurde der Pfad `/home/pi/mydb.db` angegeben. Die Einstellungen des `Logger` Nodes sind in der Abbildung 29 dargestellt.

Abbildung 29: Einstellungen des `Logger` Node `Inverter Yield`

Anschließend werden die `msg.objects` über einen `Link out` Node weitergeleitet an den `Logging Flow`. Dieser ist in der Abbildung 30 dargestellt. In dem Flow werden die `msg.objects` in den `SQL-Lite` Node übertragen und von diesem in einer `SQL-Lite` Datenbank abgespeichert. Zeitgleich werden die Datenpunkte aus dem Ringspeicher gelöscht, die ihre maximale Speicherdauer erreicht haben.

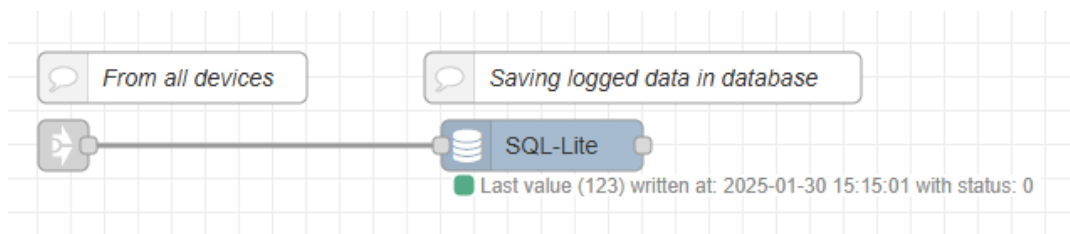
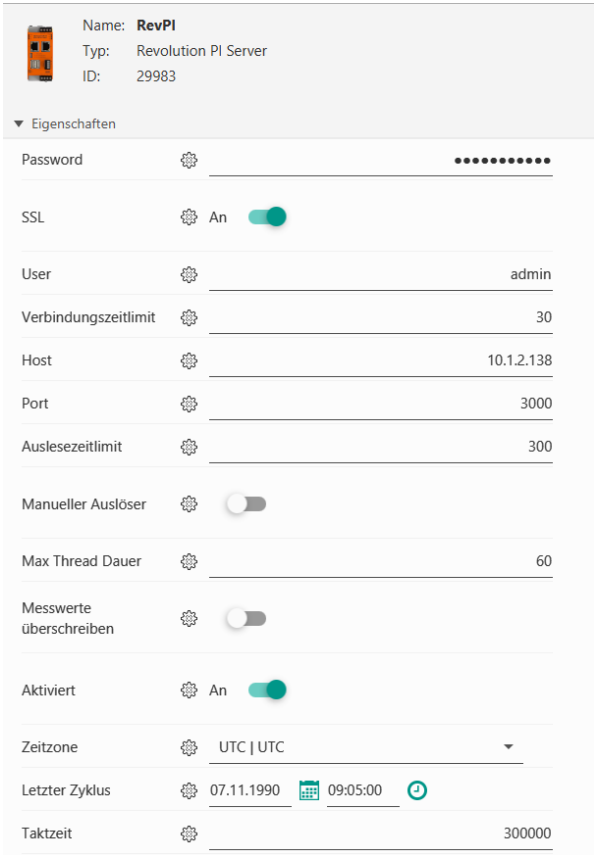


Abbildung 30: `Logging Flow`

In dem `SQL-Lite Node` muss als einzige Einstellung, die schon in den `Logger Nodes` ausgewählte Datenbank, ebenfalls eingestellt werden. In dem `Logger Flow` laufen sämtliche über Modbus abgefragten Daten der einzelnen Flows zusammen und werden im Ringspeicher des PV-Controllers abgelegt.

Aus dem Ringspeicher des PV-Controllers werden die Daten von JEVis abgefragt. Dafür wird in JEVis der RevPi, auf dem der PV-Controller implementiert wurde, als Gerät zur Datenerfassung angelegt. Das Anlegen des RevPis in JEVis ist in der Abbildung 31 dargestellt.



Name: RevPI
Typ: Revolution PI Server
ID: 29983

▼ Eigenschaften

Password	
SSL	An	<input checked="" type="checkbox"/>
User		admin
Verbindungszeitlimit		30
Host		10.1.2.138
Port		3000
Auslesezeitlimit		300
Manueller Auslöser		<input type="checkbox"/>
Max Thread Dauer		60
Messwerte überschreiben		<input type="checkbox"/>
Aktiviert	An	<input checked="" type="checkbox"/>
Zeitzone		UTC UTC ▼
Letzter Zyklus		07.11.1990 09:05:00
Taktzeit		300000

Abbildung 31: Einrichten des RevPi in JEVis

Zum Anlegen des Gerätes werden die Login-Daten sowie die Host-Adresse des RevPi benötigt. Anschließend werden noch weitere Einstellungen getroffen. Folgende Einstellungen müssen vorgenommen werden:

Verbindungszeitlimit [s]

- Nach dieser Zeit wird der Verbindungsaufbau zu RevPi gestoppt, falls dieser innerhalb dieses Zeitlimits nicht erreichbar ist.

Auslesezeitlimit [s]

- Das Auslesen der Daten wird nach dieser Zeit gestoppt unabhängig davon, ob die Daten vollständig ausgelesen wurden.

Manueller Auslöser

- Die Datenabfrage muss bei der Aktivierung dieser Funktion manuell gestartet werden.

Max Thread Dauer [s]

- Nach der eingestellten Dauer wird das Lesen eines einzelnen Datenpunktes abgebrochen.

Messwerte überschreiben

- Wenn diese Funktion aktiviert wird, werden bereits abgefragte Daten überschrieben, falls sich diese auf dem abgefragten Gerät ändern.

Aktiviert

- Durch Aktivieren dieser Einstellung wird die Datenabfrage eingeschaltet.

Zeitzone

- Hier wird die verwendete Zeitzone angegeben.

Letzter Zyklus

- Dieser gibt an wann zuletzt Daten abgefragt wurden.

Taktzeit [ms]

- Nach diesem einzustellenden Intervall werden die Daten vom RevPi angefragt. Demnach werden bei einem Abfrageintervall von 300.000 ms alle 5 Minuten die Daten des RevPis abgefragt.

Nachdem diese Einstellungen getroffen wurden, werden die Daten in JEVIs vollständig eingelesen. In JEVIs müssen die Daten noch abhängig von Ihrer `Trend ID` einer Datenreihe zugeordnet werden. Dafür müssen JEVIs die jeweiligen `Trend IDs` der Datenreihen angegeben werden. Die Zuordnung einer Datenreihe ist in der Abbildung 32 dargestellt. Die `Trend ID` wird in dem Feld `Source Id` angegeben. Die Datenreihe, in die die Werte übernommen werden, wird in dem Feld `Target Id` angegeben.

The screenshot shows the configuration page for a device named 'Hauptzähler2Energy'. At the top, there is a digital display icon showing '109'. Below it, the name 'Hauptzähler2Energy' is displayed, followed by 'Typ: Revolution PI Channel' and 'ID: 45064'. A section titled 'Eigenschaften' (Properties) contains several fields: 'Source Id' with a gear icon and the value 'fbf2f46773a02bbf'; 'Target Id' with a gear icon, a dropdown menu showing '[45062] Hauptzähler2Energy - Value', and a 'Gehe zu' (Go to) button; 'Status Log' with a gear icon and the value '16'; 'Letzte Auslesung' (Last Reading) with a gear icon, a date field showing '31.01.2025', a time field showing '10:11:00', and a clock icon; 'Versatz der Auslesung' (Reading Offset) with a gear icon and an empty field; and 'Maximale Auslesung' (Maximum Reading) with a gear icon and an empty field.

Abbildung 32: Zuordnung einer Datenreihen in JEVIs

Diese Einstellungen müssen für jeden Datenpunkt, der betrachtet werden soll, einmal angelegt werden. Bei der Einbindung weiterer Daten in JEVIs kann diese Zuordnung der Datenpunkte in JEVIs kopiert werden und muss nicht erneut vorgenommen werden.

5.5 Steuerung der PV-Anlagen

In der Regel werden die PV-Anlagen im Maximum-Power-Point (MPP) betrieben, um den maximal möglichen Ertrag zu erzielen. Aus den regulatorischen Anforderungen in Abschnitt 2.3 sind jedoch die Anforderungen entstanden, dass der Versorgungsnetzbetreiber über die FWA mit Hilfe von Steuerbefehlen eine Leistungsanpassung vornehmen kann. Ebenfalls werden zum Verhindern einer Abschaltung durch die $P_{AV,E}$ -Überwachung vor Erreichen der maximalen Einspeiseleistung die PV-Anlagen gedrosselt.

Für die Kommunikation mit der FWA wurde ein eigener Flow erstellt. Dieser heißt `Flow_FWA` und ist in der Abbildung 33 dargestellt. Für die Steuerung der PV-Anlagen werden zunächst die Steuersignale der FWA im oberen Bereich des `FWA_Flows`, abgefragt. Die Abfrage der FWA wird im `Main_Flow` gestartet und folgt dem in Abschnitt 5.3.1 Modbus-Datenabfrage beschriebenen Ablauf.

Es werden folgende Steuersignale, aus Tabelle 2, von der FWA abgefragt:

Leistungssteuerung

- Sollwertvorgabe Wirkleistung
- Name in Node-RED: `FWA power control target`

Q(P) Steuerung

- Umschaltung Blindleistungsvorgabe durch Kennlinie oder NLS der WEMAG
- Name in Node-RED: `FWA reactive power control mode`

Blindleistungssteuerung

- Sollwertvorgabe Blindleistung
- Namen in Node-RED: `FWA reactive power control target`

LS(Q0) oder LaTS

- Not-Aus-Signal
- Namen in Node-RED `FWA emergency Stop`

Das Not-Aus-Signal wird auf dem PV-Controller in der Variable `global.FWA.emergency_stop` gespeichert und den anderen Flows zur Verfügung gestellt. Dadurch liegt das Not-Aus-Signal bereits auf dem PV-Controller vor und kann in Zukunft verwendet werden, um mit dem PV-Controller die LS oder LaTS der Erzeugungsanlagen zu schalten.

Die abgefragten Steuersignale werden, bis auf das Not-Aus-Signal, anschließend an den `PV Control Flow`, zur Weiterverarbeitung übertragen. In dem `PV Control Flow` werden, die von der FWA abgefragten Steuersignale weiterverarbeitet. Der `PV Control Flow` ist in der Abbildung 34 abgebildet. Über einen `Link in Node` werden die abgefragten Steuersignale aus dem `FWA Flow` an den `PV Control Flow` übertragen. Der `PV Control Flow` benötigt ebenfalls den Netzbezug für die Unterstützung der $P_{AV,E}$ -

Überwachung. Dieser wird, falls ein Bezugszähler angeschlossen ist, dem PV Control Flow als globale Variable zugänglich gemacht. Dadurch kann der Flow unabhängig von der Abfrage des Netzbezuges ausgeführt werden. Vor der Weiterverarbeitung der Daten wird durch einen Join Node sichergestellt, dass alle Steuersignale von der FWA vorliegen und dem nachfolgende Function Node Processing of FWA control signals als ein `msg.object` übergeben werden.

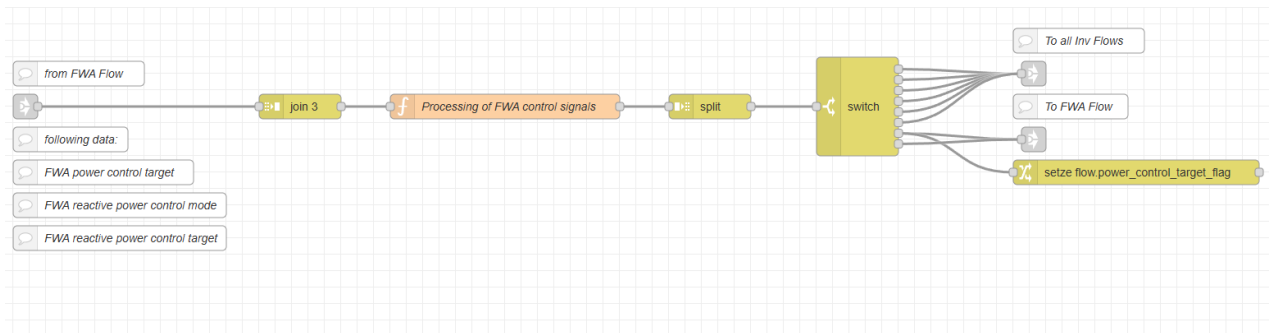


Abbildung 34: PV Control Flow

In dem Function Node Processing of FWA control signals werden die Steuersignale verarbeitet und es werden die benötigten Steuerbefehle für die Wechselrichter sowie die Rückmeldungen an die FWA erzeugt. In der Abbildung 35 und der Abbildung 36 ist der Code des Function Node Processing of FWA control signals dargestellt.


```

1  // This function is processing the control signals of the FWA and outputs inverter
2  // controls and feedback to the FWA
3
4  //declare variables
5  var grid_power_meter_enable = global.get("grid_power_meter_enable");          // Import
6  global.grid_power_meter_enable
7  var max_grid_feed_in = global.get("grid.max_grid_feed_in");                  // Import
8  global.grid.max_grid_feed_in
9  var current_grid_consumption = global.get("PM.current_grid_consumption");    // Import
10 global.PM.current_grid_consumption
11 var power_control_target_flag = flow.get("power_control_target_flag");        // Import
12 flow.power_control_target_flag used to remember last power target
13 var new_power_control_target = 0;                                           //used to save the new power target
14 to increase readability of the code
15 var new_reactive_power_control_target = 0;                                   //used to save the new reactive power
16 target to increase readability of the code
17 var newMsg = {};                                                            //new msg.object to return
18 newMsg.payload = {};                                                        //create payload object within newMsg
19 Object
20
21 // Power Control of FWA (in %)
22 //saving new power control target in variable
23 new_power_control_target = msg.payload["FWA power control target"];
24
25 //If there is no powerlimit and the Inv is not throttled. --> Disable powerlimit so
26 //no timer for the power limit needs to be refreshed.
27 //If there is no powerlimit from FWA the powerlimit can be set individually.
28 //To prevent shut off of Inverters via the Pave Controller the power will be reduced
29 //in case of to high grid feed in.
30 if (new_power_control_target == 100) {
31     //Check if grid power meter is connected and if grid consumption is higher than
32     //max_grid_feed_in minus 20kW
33     if((grid_power_meter_enable == 1) && (current_grid_consumption * (-1)) >= (
34         max_grid_feed_in - 20)){
35         // reduce power by 20%. If power reduction unsufficient power will be reduced
36         // by another 20 % next time function is run
37         power_control_target_flag = power_control_target_flag - 20;
38         if (power_control_target_flag < 0) { //limit power target to 0%
39             power_control_target_flag = 0;
40         }
41         //saving new power control target to newMsg (Output Object) --> to inverters
42         newMsg.payload["Inv power control target"] = power_control_target_flag;
43         //Enable power control. --> to inverters
44         newMsg.payload["Inv power control target Ena"] = 1;
45         //Set timeout for power control. Set to 180 s to ensure power control target
46         //will be valid until register rewrite --> to inverters
47         newMsg.payload["Inv power control target timeout"] = 180;
48         //Feedback for the FWA with current power control settings --> to FWA
49         newMsg.payload["FWA power control target feedback"] =
50             power_control_target_flag;
51     } else if ((grid_power_meter_enable == 1) && (power_control_target_flag < 100)){
52         //increase power if below max_grid_feed_in
53         power_control_target_flag = power_control_target_flag + 20;
54         if (power_control_target_flag > 100){ //limit power target to 100%
55             power_control_target_flag = 100;
56         }
57         //saving new power control target to newMsg (Output Object) --> to inverters
58         newMsg.payload["Inv power control target"] = power_control_target_flag;
59         //Enable power control. --> to inverters
60         newMsg.payload["Inv power control target Ena"] = 1;
61         //Set timeout for power control. Set to 180 s to ensure power control target
62         //will be valid until register rewrite --> to inverters
63         newMsg.payload["Inv power control target timeout"] = 180;
64         //Feedback for the FWA with current power control settings --> to FWA
65         newMsg.payload["FWA power control target feedback"] =
66             power_control_target_flag;
67     } else{
68         // Disable power control. (Inv contiunues working on 100% if not specified
69         // different) --> to inverters
70         newMsg.payload["Inv power control target Ena"] = 0;
71         //Feedback for the FWA with current power control settings --> to FWA

```

Abbildung 35: Function Node Processing of FWA control signals Teil 1

```

55         newMsg.payload["FWA power control target feedback"] = 100;
56     }
57
58 } else { //If power control target is not 100 change the inverter power control target
59     //saving new power control target to newMsg (Output Object) --> to inverters
60     newMsg.payload["Inv power control target"] = new_power_control_target;
61     //Enable power control. --> to inverters
62     newMsg.payload["Inv power control target Ena"] = 1;
63     //Set timeout for power control. Set to 180 s to ensure power control target will
        be valid until register rewrite --> to inverters
64     newMsg.payload["Inv power control target timeout"] = 180;
65     //Feedback for the FWA with current power control settings --> to FWA
66     newMsg.payload["FWA power control target feedback"] = new_power_control_target;
67 }
68
69 // Reactive Power Control of FWA(in %) and switching between control modes
70 //saving new reactive power control target in variable
71 new_reactive_power_control_target = msg.payload["FWA reactive power control target"];
72
73 //Reactive Power Control based on characteristic curve
74 if (msg.payload["FWA reactive power control mode"] == 0) {
75     //Disable reactive power control. (Inv continues working on based on the
        characteristic curve) --> to inverters
76     newMsg.payload["Inv reactive power control target Ena"] = 0;
77     //Feedback for the FWA with current reactive power control mode --> to FWA
78     newMsg.payload["FWA reactive power control mode feedback"] = 0;
79     //Reactive Power Control based on energy providers specifications
80 } else if (msg.payload["FWA reactive power control mode"] == 1) {
81     //saving new reactive power control target to newMsg (Output Object) --> to
        inverters
82     newMsg.payload["Inv reactive power control target"] =
        new_reactive_power_control_target;
83     //Enable reactive power control. --> to inverters
84     newMsg.payload["Inv reactive power control target Ena"] = 1;
85     //Set timeout for reactive power control. Set to 180 s to ensure reactive power
        control target will be valid until register rewrite --> to inverters
86     newMsg.payload["Inv reactive power control target timeout"] = 180;
87     //Feedback for the FWA with current reactive power control mode --> to FWA
88     newMsg.payload["FWA reactive power control mode feedback"] = 1;
89 }
90
91 return newMsg;
92

```

Abbildung 36: Function Node Processing of FWA control signals Teil 2

Als erstes werden die benötigten Variablen importiert und es werden Hilfs-Variablen erzeugt, um die Lesbarkeit des Codes zu verbessern. Ein neues `msg.object` mit dem Namen `newMsg` wird ebenfalls erzeugt. In diesem werden die Steuerbefehle für die Wechselrichter und die Rückmeldungen an die FWA als einzelne `payload.objects` gespeichert und als ein `msg.object` von dem Node zurückgegeben. Für jedes dieser `payload.objects` ist im Code das Zielgerät als Kommentar vermerkt. Im Folgenden wird sich zur Verbesserung der Lesbarkeit nur auf die Namen der `payload.objects` bezogen.

Anschließend wird mit der Verarbeitung der Steuersignale begonnen, angefangen mit der Wirkleistungsvorgabe. Dafür wird das von der FWA abgefragte Steuersignal `FWA power`

`control target` der Hilfs-Variable `new_power_control_target` zugewiesen. Anschließend wird die Hilfs-Variable `new_power_control_target` ausgewertet.

Beträgt diese 100 %, so kann von dem PV-Controller eine Reduzierung der Wirkleistung vorgenommen werden, um die Abschaltung durch die $P_{AV,E}$ -Überwachung zu verhindern. Diese Reduzierung wird nur vorgenommen, wenn durch die FWA keine andere Wirkleistungsvorgabe vorliegt. Auf Grund dessen, dass die Umsetzung der Steuersignale der FWA verpflichtend ist, werden sie priorisiert. Wenn nun das `new_power_control_target` 100 % beträgt und ein Bezugszähler angeschlossen ist, wird überprüft, ob die Einspeiseleistung (`current_grid_consumption`) weniger als 20 kW von der maximalen Einspeiseleistung (`max_grid_feed_in`) entfernt ist. Ist dies der Fall, so wird die Wirkleistung der Wechselrichter in 20 % Schritten reduziert bis die Einspeiseleistung wieder mehr als 20 kW von der maximalen Einspeiseleistung (`max_grid_feed_in`) entfernt ist. Die neue Wirkleistungsvorgabe für die Wechselrichter wird in der Variable `Inv power control target` abgespeichert und die äußere Wirkleistungsvorgabe der Wechselrichter wird durch Setzen von `Inv power control target Ena` zu 1 aktiviert. Durch Setzen von `Inv power control target timeout` auf 180 Sekunden wird ein Timeout für die Dauer der Wirkleistungsvorgabe eingestellt, um sicherzustellen dass die Wirkleistungsvorgabe bis zum erneuten Setzen durch den `Function Node` aktiv bleibt. Der FWA wird durch Setzen von `FWA power control target feedback` auf die Wirkleistungsvorgabe der Wechselrichter die aktuelle Wirkleistungsvorgabe mitgeteilt. Falls beim erneuten Ausführen des `Function Node` die Einspeiseleistung unterhalb der Grenze liegt, wird die Wirkleistungsvorgabe des Wechselrichters durch Setzen derselben Variablen mit der neuen Wirkleistungsvorgabe wieder um 20 % erhöht. In der Variable `power_control_target_flag` wird sich die zuletzt verwendete Wirkleistungsvorgabe gemerkt.

Falls keine Reduzierung der Wirkleistung zum Verhindern der Abschaltung durch die $P_{AV,E}$ -Überwachung benötigt wird und das `new_power_control_target` weiterhin 100 % beträgt, wird die äußere Wirkleistungsvorgabe der Wechselrichter deaktiviert. Durch das Setzen von `Inv power control target Ena` zu 0 und an die FWA wird zurückgemeldet,

dass keine Leistungsreduzierung mehr eingestellt ist durch das Setzen von `FWA power control target feedback` auf 100.

Für den Fall, dass das `new_power_control_target` nicht 100 % beträgt, wird `Inv power control target` auf den Wert von `new_power_control_target` gesetzt und die äußere Wirkleistungsvorgabe der Wechselrichter wird durch das Setzen von `Inv power control target Ena` zu 1 wieder aktiviert. Es wird ebenfalls ein Timeout für die Dauer der Wirkleistungsvorgabe durch Setzen von `Inv power control target timeout` auf 180 Sekunden eingestellt. Zuletzt wird noch die Rückmeldung der Wirkleistungsvorgabe an die FWA durch Setzen von `FWA power control target feedback` zu dem Wert von `new_power_control_target` erzeugt.

Nach der Verarbeitung der Wirkleistungsvorgabe wird die Blindleistungsvorgabe von der FWA umgesetzt. Zunächst wird das von der FWA abgefragte Steuersignal `FWA reactive power control target` der Hilfs-Variable `new_reactive_power_control_target` zugewiesen. Anschließend wird der von der FWA vorgegebene Blindleistungssteuerungsmodus `FWA reactive power control mode` ausgewertet. Wenn `FWA reactive power control mode` gleich 0 ist, soll der Wechselrichter nach der von der WEMAG vorgegebenen Blindleistungskurve arbeiten. Dafür wird die äußere Blindleistungsvorgabe durch das Setzen von `Inv reactive power control Ena` zu 0 deaktiviert. Bei der Blindleistungsvorgabe durch die FWA muss der FWA lediglich der Blindleistungssteuerungsmodus rückgemeldet werden, dafür wird `FWA reactive power control mode feedback` ebenfalls zu 0 gesetzt.

Wenn der von der FWA übertragene `FWA reactive power control mode` 1 beträgt, ist die Blindleistung nach Vorgaben der WEMAG in % bezogen auf die installierte Leistung zu regeln. Dafür wird `Inv reactive power control target` zu `new_reactive_power_control_target` gesetzt und äußere Blindleistungsvorgabe des Wechselrichters wird durch das Setzen von `Inv reactive power control target Ena` auf 1 aktiviert. Anschließend wird der Timeout der Blindleistungsvorgabe `Inv reactive power control target timeout` ebenfalls auf 180 Sekunden eingestellt und zur

Rückmeldung des Blindleistungssteuerungsmodus wird `FWA reactive power control mode feedback` zu 1 gesetzt.

Nachdem die von der FWA abgefragten Steuersignale vollständig verarbeitet wurden, werden die gesetzten Variablen in dem `msg.object newMsg` von dem Function Node zurückgegeben. Anschließend wird das `msg.object` in mehrere einzelne `msg.objects` aufgeteilt von dem auf dem Function Node folgenden Split Node, siehe Abbildung 34. Dabei wird für jedes Schlüssel/Wert-Paar ein eigenes `msg.object` erzeugt. Diese werden anschließend über ein Switch Node an alle Inverter Flows und den FWA Flow weitergeleitet zur Übertragung über Modbus an die Geräte. Das Senden der Daten wurde bereits in dem Abschnitt 5.3.2 Modbus Daten senden beschreiben. Dabei wird auch die Hilfs-Variable `power_control_target_flag` mit der aktuellen Wirkleistungsvorgabe gesetzt.

5.6 Leistungsreduzierung Verbraucher

Um zu bestimmen, ob eine Lastspitze droht und eine Leistungsreduzierung der steuerbaren Verbraucher vorgenommen werden muss, wird der kontinuierlich über Modbus abgefragte Netzbezug weiterverarbeitet. Die Voraussetzung dafür ist, dass ein Stromzähler als Bezugszähler in der Konfiguration an den PV-Controller übergeben wird, andernfalls kann von dem PV-Controller keine Empfehlung zur Leistungsreduzierung ausgegeben werden. Der Netzbezug wird dem `Power reduction Flow` übergeben, dieser ist in der Abbildung 37 dargestellt.

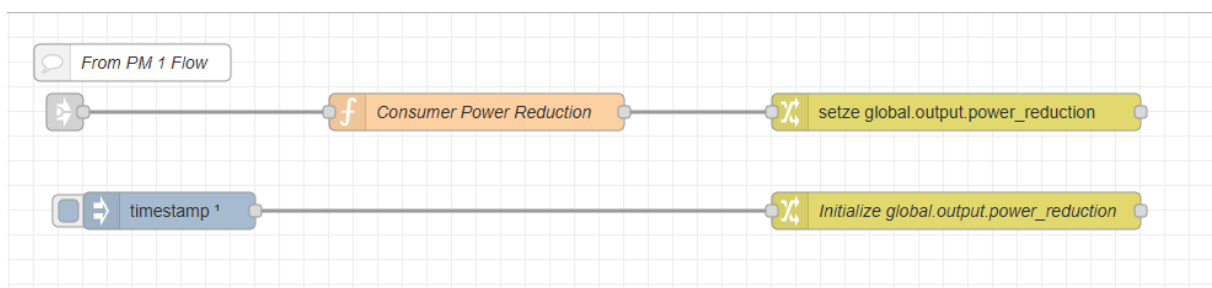


Abbildung 37: Power reduction Flow

Beim Programmstart wird die Merker-Variable `global.output.power_reduction` erzeugt. Mit Hilfe dieser Merker-Variable wird das Ergebnis des Flows gespeichert und kann bei einer späteren Erweiterung des PV-Controllers direkt auf diesem weiterverarbeitet werden. Sobald ein neuer Messwert des Netzbezuges vorliegt, wird dieser in dem Function Node Consumer Power Reduction mit der in der Konfiguration eingestellten Grenze zur Leistungsreduzierung (`global.grid.limit_power_reduction`) verglichen. Wenn die Grenze erreicht ist, gibt der Function Node eine 1 aus, unterhalb dieser Grenze wird eine 0 ausgegeben. Der Code des Function Node Consumer Power Reduction ist in der Abbildung 38 dargestellt.

```

1 //This function checks if the current Grid consumption is greater or equal to the pre set limit_power_reduction
2 //When limit_power_reduction is reached the power consumption needs to be reduced to avoid a power peak.
3
4 var limit_power_reduction = global.get("grid.limit_power_reduction") // Import global.grid.limit_power_reduction
5 var newMsg = {}; //create Object for return
6
7 if (msg.payload >= limit_power_reduction) { //Check if limit_power_reduction is reached
8     newMsg = { payload: 1 }; //Power limit reached therefore return 1 to signal needed power reduction
9 }else{
10     newMsg = { payload: 0 }; //Power limit not reached therefore return 0. No power reduction needed.
11 }
12
13 return newMsg;
```

Abbildung 38: Function Node Consumer Power Reduction

Bei der Grenze zur Leistungsreduzierung handelt es sich um eine Grenze, welche unterhalb des maximalen Netzbezuges liegt und in der Konfiguration frei definiert werden kann. Bei Überschreiten der Grenze zur Leistungsreduzierung zeigt der PV-Controller dies an, bevor der maximale Netzbezug erreicht ist und es zu einer etwaigen Spitzenlast kommt. Dadurch kann frühzeitig eine Leistungsreduzierung vorgenommen werden. Voraussetzung dafür ist jedoch, dass die Merker-Variable `global.output.power_reduction` auf dem PV-Controller in Zukunft weiterverarbeitet wird oder über z.B. OPC UA anderen Controllern zugänglich gemacht wird.

5.7 Klonen des Programms auf Kundengeräte und anderen Controller

Zum Klonen des Programms auf die Kundengeräte muss zunächst Node-RED installiert sein. Dabei ist darauf zu achten, dass Projekte in Node-RED aktiviert sind, anderenfalls ist ein Klonen über das Github Repository nicht möglich. Sobald Node-RED installiert ist, muss ein neuer SSH-Schlüssel für den Zugriff auf den Github erzeugt werden. Beim Einrichten des Public-Keys auf dem Github ist darauf zu achten, diesem nur Lesezugriff zu gewähren, um ungewollte Änderungen auf dem Github zu verhindern.

Anschließend muss ein neues Projekt erstellt werden. Dabei öffnet sich der in Abbildung 39 gezeigte Fenster.

Projekte

Projekt öffnen

Projekt erstellen

Repository lokal klonen

Projektname

Migration_neue_Plattform

Darf nur A-Z 0-9 _ enthalten

Git-Repository-URL

git@github.com:MU-Envidatec/Bachelorarbeit_MU.git

https://, ssh:// oder file://

SSH-Schlüssel

SSH_Key_Rev_Pi

Passphrase

.....

Schlüssel für Credentials

.....

Abbrechen

Repository lokal klonen

Abbildung 39: Git Repository klonen auf Kundengeräte

Durch die Auswahl von Repository lokal klonen, kann das Programm vom Github heruntergeladen werden. Dafür muss zunächst ein Projektname vergeben werden. Anschließend muss die URL des Git Repository angegeben werden. Es muss der neue erstellte SSH-Schlüssel ausgewählt werden und falls eine Passphrase verwendet wird, muss diese ebenfalls angegeben werden. Zuletzt muss noch der Schlüssel für Credentials eingegeben werden. Bei den Credentials handelt es sich um Eigenschaften der Nodes, welche möglicherweise vertrauliche Informationen beinhalten. Diese Eigenschaften werden nicht zusammen mit dem eigentlichen Node-RED Flow gespeichert, sondern in einer eigenen Credentials-Datei [51]. Beim Erstellen des Projekts wurde ein Schlüssel für diese vergeben. Dieser muss ebenfalls eingetragen werden. Nun kann über die Schaltfläche „Repository lokal klonen“ das Programm auf dem Kundengerät installiert werden.

Der verwendete Revolution Pi Core SE könnte in Zukunft nicht mehr verfügbar sein oder durch ein nachfolge Modell ersetzt werden. Um den PV-Controller dennoch weiter verwenden zu können, kann dieser einfach auf einer anderen Plattform, die Node-RED unterstützt, geklont werden. Die Schritte dafür sind identisch zum Klonen auf Kundengeräte. Es muss lediglich Schreibzugriff auf den Github gewährt werden. Möglich ist dies dadurch, dass keine herstellerbezogenen Nodes bzw. Paletten in der Programmierung verwendet wurden, wodurch keine Anpassungen an dem Programm vorgenommen werden müssen.

6 Excel-Konfigurationstool

Um eine einfache Konfiguration des PV-Controllers zu ermöglichen, wird ein Konfigurationstool in Excel erstellt. Ziel des Konfigurationstools ist es, nach dem vollständigen Ausfüllen des Tools eine csv-Konfigurationsdatei zu erstellen mit der die Konfiguration des PV-Controllers und der angeschlossenen Geräte vorgenommen werden kann. Nach dem Erstellen der Konfiguration kann die csv-Konfigurationsdatei auf den PV-Controller übertragen werden und wird selbstständig von dem PV-Controller bei Programmstart eingelesen. Das Konfigurationstool wird so entworfen, dass es um weitere Funktionen erweitert werden kann. In den folgenden Abschnitten wird der Aufbau des Konfigurationstools beschrieben.

Die Übertragung der Konfigurationsdatei und der händischen Konfiguration auf den PV-Controller wird in dem Konfigurationstool übersichtlich beschrieben. Durch die größtenteils csv gestützte Konfiguration kann der PV-Controller auch von programmierunerfahrenen Personen mit Hilfe der geführten Konfiguration des Exceltool konfiguriert werden.

Zur Verbesserung der Lesbarkeit wird, wenn sich auf Inhalte oder Funktionen des Excel-Konfigurationstool bezogen wird, dies ebenfalls in der Schriftart `Courier New` dargestellt.

Das Excel-Konfigurationstool ist in Anhang F zu finden.

6.1 Allgemeiner Aufbau

Das Konfigurationstool ist auf mehrere Tabellenblätter aufgeteilt und wird sequenziell durchgearbeitet. Am Ende der Tabellenblätter befinden sich Navigationsknöpfe, mit denen sich in der richtigen Reihenfolge durch das Konfigurationstool navigiert werden kann. Zusätzlich befindet sich am linken Rand jedes Tabellenblattes eine Kapitelauswahl mit der frei zwischen den Tabellenblättern gewechselt werden kann. In einigen Feldern sind die möglichen Angaben in einer Dropdownliste vorgegeben, um die Eingabe zu erleichtern

und fehlerhafte Eingaben zu verhindern. Die Inhalte der Dropdownlisten werden in dem Tabellenblatt `Listen` und `Hinweise` verwaltet.

In dem Tabellenblatt `Konfiguration` wird die Konfigurationsdatei aus den unterschiedlichen Nutzereingaben zusammengesetzt. Dafür werden die für die Konfigurationsdatei relevante Daten aus den Tabellenblättern mit Hilfe der `ÜBERNEHMEN` Funktion in Excel in das Tabellenblatt `Konfiguration` übertragen.

Für das Konfigurationstool wurden Designregeln festgelegt. Zur Verbesserung der Lesbarkeit der eingestellten Formeln werden einzelne Zellen, die in den Formeln verwendet werden, über den Namenmanager mit einem aussagekräftigen Bezeichner versehen. Dies hat zur Folge, dass die Übersichtlichkeit der Formeln steigt und diese einfacher nachzuvollziehen sind. Tabellenblätter, welche keine Eingaben durch den Nutzer erfordern und nur für die Funktion des Konfigurationstools benötigt werden, werden durch ein Passwort geschützt, sodass keine ungewollten Änderungen auf diesen Tabellenblättern vorgenommen werden können.

Die Modbus-Registerlisten sind für jeden Gerätetyp in einem eigenen Tabellenblatt hinterlegt. Es existieren folgende Tabellenblätter mit Modbus-Registerlisten:

`WR_Modbusregisterlisten`

- Für die Wechselrichter.

`PM_Modbusregisterlisten`

- Für die Stromzähler.

`FWA_Modbusregisterlisten`

- Für die FWA.

`Pyr_Modbusregisterlisten`

- Für die Pyranometer

In den folgenden Abschnitten werden die einzelnen vom Nutzer auszufüllenden Tabellenblätter erläutert.

6.2 Deckblatt

Das Konfigurationstool beginnt auf dem Tabellenblatt `Deckblatt`, dieses ist in der Abbildung 40 dargestellt. Auf dem Tabellenblatt wird kurz beschrieben, wofür das Konfigurationstool genutzt wird. Über das Anklicken der Zelle `Konfiguration Starten` wird das nächste Tabellenblatt `Projektdaten` geöffnet und es kann mit der Konfiguration begonnen werden.

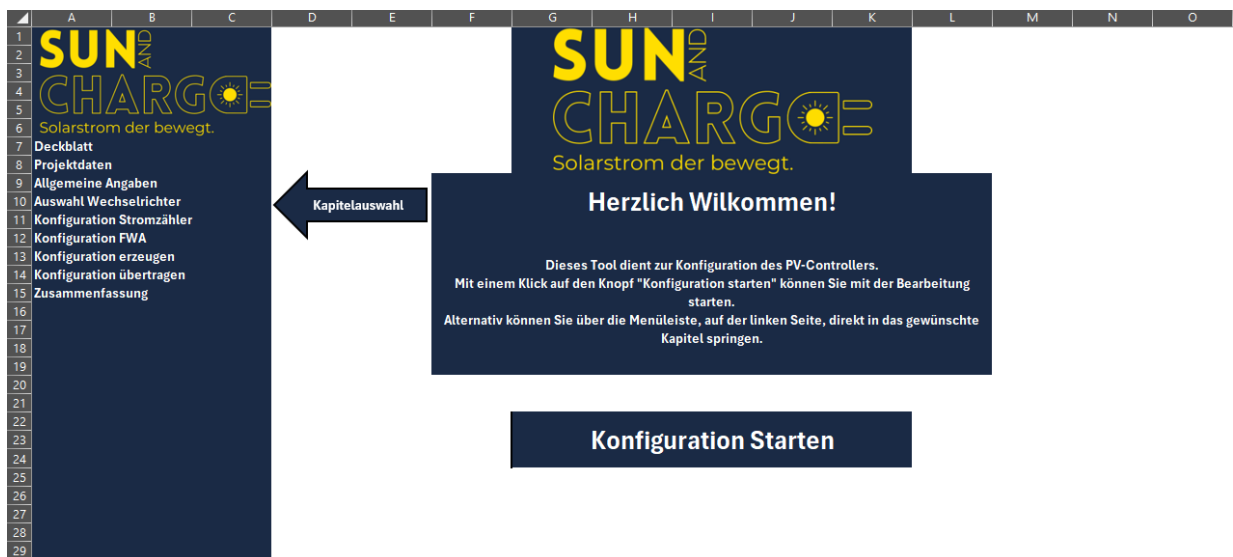



Abbildung 40: Konfigurationstool Tabellenblatt: Deckblatt

6.3 Projektdaten

In dem Tabellenblatt `Projektdaten`, siehe Abbildung 41, werden zunächst die allgemeinen Projektdaten abgefragt. Dafür müssen die Projektnummer, sowie Kundendaten (Auftraggeber, Adresse, Kontakt und Leistungsort) eingegeben werden. Es wird ebenfalls der Projektzeitraum und das geplante Budget abgefragt. Unter der Abfrage der Projektdaten befindet sich ein Freitextfeld für Anmerkungen zum Projekt. Alle auf diesem Tabellenblatt eingegebenen Daten sind jedoch nicht relevant für die eigentliche Konfiguration des PV-

Controllers. Sie helfen jedoch die entstandene Konfiguration einem Kunden bzw. Projekt zuzuordnen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35														
36														
37														
38														
39														
40														
41														
42														
43														
44														
45														
46														
47														
48														
49														
50														
51														
52														

Projektdaten

Projektname / -Nr. _____

Auftraggeber: _____

Firmenadresse: _____

Straße: _____

PLZ: _____

Ort: _____

Leistungsort: _____

Straße: _____

PLZ: _____

Ort: _____

Kontakt

	Technisch	Kaufmännisch
Name:	_____	_____
Email:	_____	_____
Telefon:	_____	_____

Projektstart: _____

Projektende (geplant): _____

Budget (geplant): _____

Weiter Anmerkungen:

Zurück

Weiter

Abbildung 41: Konfigurationstool Tabellenblatt: Projektdaten


6.4 Allgemeine Angaben

Nach der Abfrage der Projektdaten folgt das Tabellenblatt `Allgemeine Angaben`. Wie in der Abbildung 42 zu erkennen, werden zunächst Angaben zum Netz und Netzanschluss abgefragt. Dafür werden folgende Informationen abgefragt:

- Spannungsebene des Netzanschlusses
- Maximaler Netzbezug
- Maximale Einspeiseleistung
- Grenze Leistungsreduzierung
- Versorgungsnetzbetreiber inclusive der Kontaktdaten

Die maximale Einspeiseleistung wird verwendet, um die Erzeugungsleistung der PV-Anlagen frühzeitig zu reduzieren um eine Abschaltung durch die $P_{AV,E}$ -Überwachung zu verhindern.

Die Grenze der Leistungsreduzierung wird benötigt, um für steuerbare Verbrauchseinrichtungen eine Empfehlung zur Leistungsreduzierung auszugeben zu können. Diese Grenze kann frei gewählt werden, sie muss jedoch unterhalb des maximalen Netzbezuges liegen. Wenn dies nicht der Fall ist, wird durch oranges Einfärben der Zelle signalisiert, dass die Eingabe fehlerhaft ist.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1																
2																
3																
4																
5																
6																
7	Solarstrom der bewegt.															
8	Deckblatt															
9	Projektdatei															
10	Allgemeine Angaben															
11	Auswahl Wechselrichter															
12	Konfiguration Stromzähler															
13	Konfiguration FWA															
14	Konfiguration erzeugen															
15	Konfiguration übertragen															
16	Zusammenfassung															
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																
49																
50																
51																
52																

Allgemeine Angaben

Allgemeine Angaben Netz:

Spannungsebene Netzanschluss:	Niederspannung
Maximaler Netzbezug:	1200 in kW
Maximale Einspeiseleistung:	270 in kW
Grenze Leistungsreduzierung:	1500 in kW
Versorgungsnetzbetreiber:	WEMAG
Kontakt Versorgungsnetzbetreiber:	Name:
	Email:
	Telefon:

Weitere Erzeugungsanlagen:

Existieren weitere Erzeugungsanlagen hinter dem selben Netzanschlusspunkt?	Ja
Existiert bereits ein Netzsicherheitsmanagement?	Ja
Art der Anlage:	Fernwirkanlage
Welche Erzeugungsanlagen befinden sich vor Ort?	
Solaranlagen:	Anzahl: 2 in Stk.
	Gesamte Leistung: 160 in kWp
BHKW:	Anzahl: 0 in Stk.
	Gesamte Leistung: in kW
Windkraftanlagen:	Anzahl: 0 in Stk.
	Gesamte Leistung: in kW

Zurück
Weiter

Abbildung 42: Konfigurationstool Tabellenblatt: Allgemeine Angaben

Im Anschluss wird abgefragt, ob sich weitere Erzeugungsanlagen hinter dem NAP befinden. Dafür wird die Art der Erzeugungsanlage sowie die Nennleistungen abgefragt. Diese Angaben werden benötigt, um die gesamte installierte Erzeugungsleistung hinter dem NAP zu erfassen. Es wird ebenfalls abgefragt, ob bereits ein Netzsicherheitsmanagement existiert und ob es sich um eine FWA oder einen FRSE handelt, die für das Netzsicherheitsmanagement verwendet wird.

6.5 Auswahl Wechselrichter

Anschließend werden die verbauten Wechselrichter auf dem Tabellenblatt `Auswahl Wechselrichter` konfiguriert. Bei der Auswahl der Wechselrichter wird zunächst die Anzahl an verwendeten Wechselrichter abgefragt. Aktuell ist es möglich, bis zu sechs Wechselrichter an den PV-Controller anzuschließen. Diese Anzahl kann in Zukunft durch geringen programmieraufwand erhöht werden. Abhängig von der Anzahl an Wechselrichtern wird in der darunterliegenden Wechselrichterkonfiguration in der Zelle `Aktiv` durch eine grüne oder rote Färbung signalisiert, welche Wechselrichter konfiguriert werden müssen. In der Abbildung 43 ist das Tabellenblatt `Auswahl Wechselrichter` dargestellt. Hier ist gut zu erkennen, welche der Wechselrichter aktiv sind.

The screenshot shows the 'Auswahl Wechselrichter' (Select Inverter) configuration window. It displays six inverters, each with a set of configuration parameters. The 'Aktiv' (Active) status is indicated by a green background for Inverter 1 and red for the others. A warning box on the right provides instructions for RS-485 to TCP/IP conversion.

Auswahl Wechselrichter	
Anzahl Wechselrichter:	1
Ringspeicher:	60 in Tagen

Wechselrichter 1	
Aktiv	TCP
Modbus Typ	TCP
IP-Adresse	10.1.2.100
Port	502
Unit-ID	1
Hersteller	SMA
Modell	STP 110-60
Nennleistung [kW]	110

Wechselrichter 2	
Aktiv	RTU
Modbus Typ	RTU
IP-Adresse	10.1.2.101
Port	502
Unit-ID	2
Hersteller	SMA
Modell	STP 110-60
Nennleistung [kW]	110

Hinweise:

Bei der Inbetriebnahme der Wechselrichter muss die Modbusschnittstelle aktiviert werden. Dabei wird auch die IP-Adresse und der Port des Wechselrichters festgelegt.

Zur Anbindung der Geräte die eine Modbus RTU Verbindung verwenden wird ein RS-485 zu TCP/IP Umsetzer benötigt!

Wechselrichter 3	
Aktiv	TCP
Modbus Typ	TCP
IP-Adresse	10.1.2.102
Port	502
Unit-ID	3
Hersteller	SMA
Modell	STP 110-60
Nennleistung [kW]	110

Wechselrichter 4	
Aktiv	TCP
Modbus Typ	TCP
IP-Adresse	10.1.2.103
Port	502
Unit-ID	4
Hersteller	SMA
Modell	STP 110-60
Nennleistung [kW]	110

Wechselrichter 5	
Aktiv	TCP
Modbus Typ	TCP
IP-Adresse	10.1.2.103
Port	502
Unit-ID	5
Hersteller	SMA
Modell	STP 110-60
Nennleistung [kW]	110

Wechselrichter 6	
Aktiv	TCP
Modbus Typ	TCP
IP-Adresse	10.1.2.104
Port	502
Unit-ID	6
Hersteller	SMA
Modell	STP 110-60
Nennleistung [kW]	110

Zurück
Weiter

Abbildung 43: Konfigurationstool Tabellenblatt: Auswahl Wechselrichter

Für alle Wechselrichter kann anschließend die Speicherdauer, der von den Wechselrichtern abgefragten Daten, im Ringspeicher des PV-Controllers ausgewählt werden. Dabei

sind Speicherdauern von bis zu einem Jahr möglich. Für die anderen Geräte lässt sich ebenfalls eine Speicherdauer von bis zu einem Jahr einstellen.

Nun können die einzelnen Wechselrichter konfiguriert werden. Zunächst wird der Modbus-Typ für den Anschluss des Wechselrichters ausgewählt. Dabei kann zwischen Modbus TCP und RTU ausgewählt werden. Falls eine Modbus-RTU-Verbindung ausgewählt wurde, wird in den Hinweisen auf der rechten Seite des Tabellenblatts der Hinweis eingeblendet, dass ein RS-485 zu TCP/IP Umsetzer benötigt wird, um den Wechselrichter an den PV-Controller anzuschließen. Anschließend müssen die IP-Adresse und der Port des Wechselrichters bzw. des RS-485 zu TCP/IP Umsetzer eingetragen werden. Falls eine Modbus-RTU-Verbindung verwendet wird, muss ebenfalls eine Unit ID angegeben werden. Es wird zusätzlich darauf hingewiesen, dass die IP-Adresse und der Port auf dem jeweiligen Wechselrichter festgelegt werden.

Zuletzt muss für jeden angeschlossenen Wechselrichter der Hersteller und das Modell ausgewählt werden. Abhängig von dem ausgewählten Wechselrichter werden die Modbus-Registerlisten in die Konfigurationsdatei übernommen. Aktuell steht nur der SUNNY TRIPOWER CORE 2 (STP 110-60) von SMA zu Auswahl.

Die Modbus-Registerlisten der Wechselrichter sind in dem Tabellenblatt `WR_Modbusregisterlisten` hinterlegt, ein Ausschnitt dessen ist in der Abbildung 44 dargestellt.

	A	B	C	D	E	F
1	Kein Wechselrichter				SMA STP 110-60	
2			R/W	Bezeichner		
3	Inv unit id	0			Inv unit id	2
4	Inv enable	0			Inv enable	1
5	Inv read fc	0			Inv read fc	3
6	Inv write fc	0			Inv write fc	6
7	Inv Amps add	0 r	AC Current		Inv Amps add	40073
8	Inv Amps qty	0 r			Inv Amps qty	1
9	Inv Amps Phase A add	0 r	Phase A Current		Inv Amps Phase A add	40074
10	Inv Amps Phase A qty	0 r			Inv Amps Phase A qty	1
11	Inv Amps Phase B add	0 r	Phase B Current		Inv Amps Phase B add	40075
12	Inv Amps Phase B qty	0 r			Inv Amps Phase B qty	1
13	Inv Amps Phase C add	0 r	Phase C Current		Inv Amps Phase C add	40076
14	Inv Amps Phase C qty	0 r			Inv Amps Phase C qty	1
15	Inv Amps SF add	0 r			Inv Amps SF add	40077
16	Inv Amps SF qty	0 r			Inv Amps SF qty	1
17	Inv Phase Voltage AN add	0 r	Phase Voltage AN		Inv Phase Voltage AN add	40081
18	Inv Phase Voltage AN qty	0 r			Inv Phase Voltage AN qty	1
19	Inv Phase Voltage BN add	0 r	Phase Voltage BN		Inv Phase Voltage BN add	40082
20	Inv Phase Voltage BN qty	0 r			Inv Phase Voltage BN qty	1
21	Inv Phase Voltage CN add	0 r	Phase Voltage CN		Inv Phase Voltage CN add	40083
22	Inv Phase Voltage CN qty	0 r			Inv Phase Voltage CN qty	1
23	Inv Voltage SF add	0 r			Inv Voltage SF add	40084
24	Inv Voltage SF qty	0 r			Inv Voltage SF qty	1
25	Inv Watts add	0 r	AC Power		Inv Watts add	40085
26	Inv Watts qty	0 r			Inv Watts qty	1
27	Inv Watts SF add	0 r			Inv Watts SF add	40086
28	Inv Watts SF qty	0 r			Inv Watts SF qty	1
29	Inv Hz add	0 r	Line Frequency		Inv Hz add	40087
30	Inv Hz qty	0 r			Inv Hz qty	1
31	Inv Hz SF add	0 r			Inv Hz SF add	40088
32	Inv Hz qty	0 r			Inv Hz qty	1
33	Inv VA add	0 r	AC Apparent Power		Inv VA add	40089
34	Inv VA qty	0 r			Inv VA qty	1
35	Inv VA SF add	0 r			Inv VA SF add	40090
36	Inv VA SF qty	0 r			Inv VA SF qty	1
37	Inv Var add	0 r	AC Reactive Power		Inv Var add	40091
38	Inv Var qty	0 r			Inv Var qty	1
39	Inv Var SF add	0 r			Inv Var SF add	40092
40	Inv Var SF qty	0 r			Inv Var SF qty	1
41	Inv PF add	0 r	AC Power Factor		Inv PF add	40093
42	Inv PF qty	0 r			Inv PF qty	1
43	Inv PF SF add	0 r			Inv PF SF add	40094
44	Inv PF SF qty	0 r			Inv PF SF qty	1
45	Inv WattHours add	0 r	AC Energy		Inv WattHours add	40095
46	Inv WattHours qty	0 r			Inv WattHours qty	2
47	Inv WattHours SF add	0 r			Inv WattHours SF add	40097
48	Inv WattHours SF qty	0 r			Inv WattHours SF qty	1
49	Inv Operating State add	0 r	Enumerated value. Oper		Inv Operating State add	40109
50	Inv Operating State qty	0 r			Inv Operating State qty	1

Abbildung 44: Ausschnitt Konfigurationstool Tabellenblatt: WR_Modbusregisterlisten

Abhängig von dem ausgewählten Wechselrichtermodell wird mit Hilfe der Übernehmen-Funktion aus Excel die zugehörige Registerliste für jeden Wechselrichter in das Tabellenblatt Konfiguration übernommen. Wenn kein Wechselrichter angeschlossen ist, wird ein eine ausschließlich mit 0 ausgefüllte Modbus-Registerliste in das Tabellenblatt Konfiguration übernommen. Diese ist in der Abbildung 44 auf der linken Bildhälfte zu

sehen. Dadurch, dass `Inv enable` in der Modbus-Registerliste auf 0 gesetzt wird, weiß der PV-Controller, dass dieser Wechselrichter nicht konfiguriert ist.

Für die aktiven Wechselrichter muss noch die Nennleistung eingetragen werden. Anschließend ist die Auswahl der Wechselrichter abgeschlossen und die Konfiguration kann auf dem nächsten Tabellenblatt fortgeführt werden.

6.6 Konfiguration Stromzähler

Im Anschluss können auf dem Tabellenblatt `Konfiguration Stromzähler` die angeschlossenen Stromzähler konfiguriert werden, das ist in der Abbildung 45 dargestellt. Dafür muss zunächst die Anzahl der Stromzähler angegeben werden. Wie auch schon bei der Auswahl der Wechselrichter, wird durch die Färbung der Zelle `Aktiv` signalisiert, welcher Stromzähler konfiguriert werden muss. Ebenfalls wird die Speicherdauer der Daten im Ringspeicher festgelegt. Des Weiteren wird abgefragt, ob der Stromzähler 1 als Bezugszähler verwendet wird. Wenn kein Bezugszähler konfiguriert wurde, kann der PV-Controller keine Messdaten des NAP an die FWA übertragen und es kann keine Leistungsreduzierung für Verbraucher angezeigt werden. Wenn kein Bezugszähler konfiguriert ist, werden zwei Hinweise eingeblendet, die auf die Einschränkung der Funktion des PV-Controllers hinweisen.

Im Anschluss werden die einzelnen Stromzähler konfiguriert. Dafür muss zunächst ein Stromzählermodell ausgewählt werden. Bisher ist nur der Janitza MID-Energiezähler B24 Stromzähler im Konfigurationstool gepflegt. Falls dieser eine Modbus-RTU-Verbindung verwendet, wird der Hinweis eingeblendet, dass ein RS-485 zu TCP/IP Umsetzer benötigt wird. Eine Unit-ID muss in diesem Fall ebenfalls angegeben werden. Diese wird direkt auf dem Stromzähler festgelegt. Anschließend müssen noch die IP-Adresse und der Port des Stromzählers bzw. des RS-485 zu TCP/IP Umsetzer angegeben werden. Die Modbus-Registerlisten der Stromzähler sind in dem Tabellenblatt `PM_Modbusregisterlisten` abgespeichert. Abhängig von den ausgewählten Zählern werden die jeweiligen Modbus-Registerlisten der Stromzähler in das Tabellenblatt `Konfiguration` übernommen.

6.7 Konfiguration FWA

Als nächstes wird auf dem Tabellenblatt `Konfiguration FWA` die FWA konfiguriert. Dabei wird auch zeitgleich das für die FWA benötigte Pyranometer mitkonfiguriert. Das Tabellenblatt `Konfiguration FWA` ist in der Abbildung 46 dargestellt.

Zu Beginn wird abgefragt, ob eine FWA vorhanden ist. Falls dies nicht der Fall ist, werden alle grün gefärbten Zellen rot und signalisieren, dass keine weiteren Eingaben erforderlich sind. Der PV-Controller arbeitet dann ohne eine FWA und nimmt keine Steuereingriffe an den Wechselrichtern vor. Dies kann dazu führen, dass wenn von keinem anderen Gerät Steuereingriffe an den Wechselrichtern vorgenommen werden, die Anforderungen an das Netzsicherheitsmanagement nicht erfüllt werden. Falls keine FWA konfiguriert wurde, wird bei den Hinweisen auf der rechten Seite des Tabellenblatts darauf hingewiesen.

Wenn eine FWA konfiguriert wird, ist zunächst die Speicherdauer des Ringspeichers auszuwählen und anschließend die IP-Adresse und der Port der FWA anzugeben. Anschließend werden die Adressen der Modbus-Register der FWA abgefragt. Diese sind nicht einheitlich definiert und werden bei der Programmierung der FWA festgelegt. Diese festgelegten Adressen sind in das Konfigurationstool zu übertragen. Diese eingegebene Konfiguration der FWA wird in das Tabellenblatt `Konfiguration` übernommen. Falls keine FWA angeschlossen ist, wird wie auch schon bei den anderen Geräten eine mit 0 gefüllte Modbus-Registerliste übernommen. Diese mit 0 gefüllte Modbus-Registerliste ist in dem Tabellenblatt `FWA_Modbusregisterlisten` hinterlegt.

Nach der FWA kann, falls benötigt, noch ein Pyranometer konfiguriert werden. Dafür ist das Modell sowie die Speicherdauer im Ringspeicher auszuwählen. Bei dem bisher implementierten Pyranometer wird ein RS-485 zu TCP/IP Umsetzer benötigt. Darauf wird bei der Auswahl des Pyranometers hingewiesen. Zuletzt ist die IP-Adresse und der Port des Pyranometers bzw. des RS-485 zu TCP/IP Umsetzer anzugeben. Die Modbus-Registerlisten der Pyranometer sind in dem Tabellenblatt `Pyr_Modbusregisterlisten` hinterlegt und werden von dort übernommen.

6.8 Konfigurationsdatei erzeugen

Nach dem Ausfüllen der vorherigen Tabellenblätter wird in dem Tabellenblatt `Konfigurationsdatei erzeugen` die Konfigurationsdatei erzeugt. Zunächst wird darauf hingewiesen, dass die Konfiguration als csv-Datei ausgegeben wird und diese im selben Verzeichnis wie das Konfigurationstool gespeichert wird. Durch das Klicken auf die Schaltfläche `Konfiguration erzeugen` wird die csv-Konfigurationsdatei automatisch erzeugt und ein Systemdialog informiert über den erfolgreichen Export. Das Tabellenblatt `Konfigurationsdatei erzeugen` ist in der Abbildung 47 abgebildet.

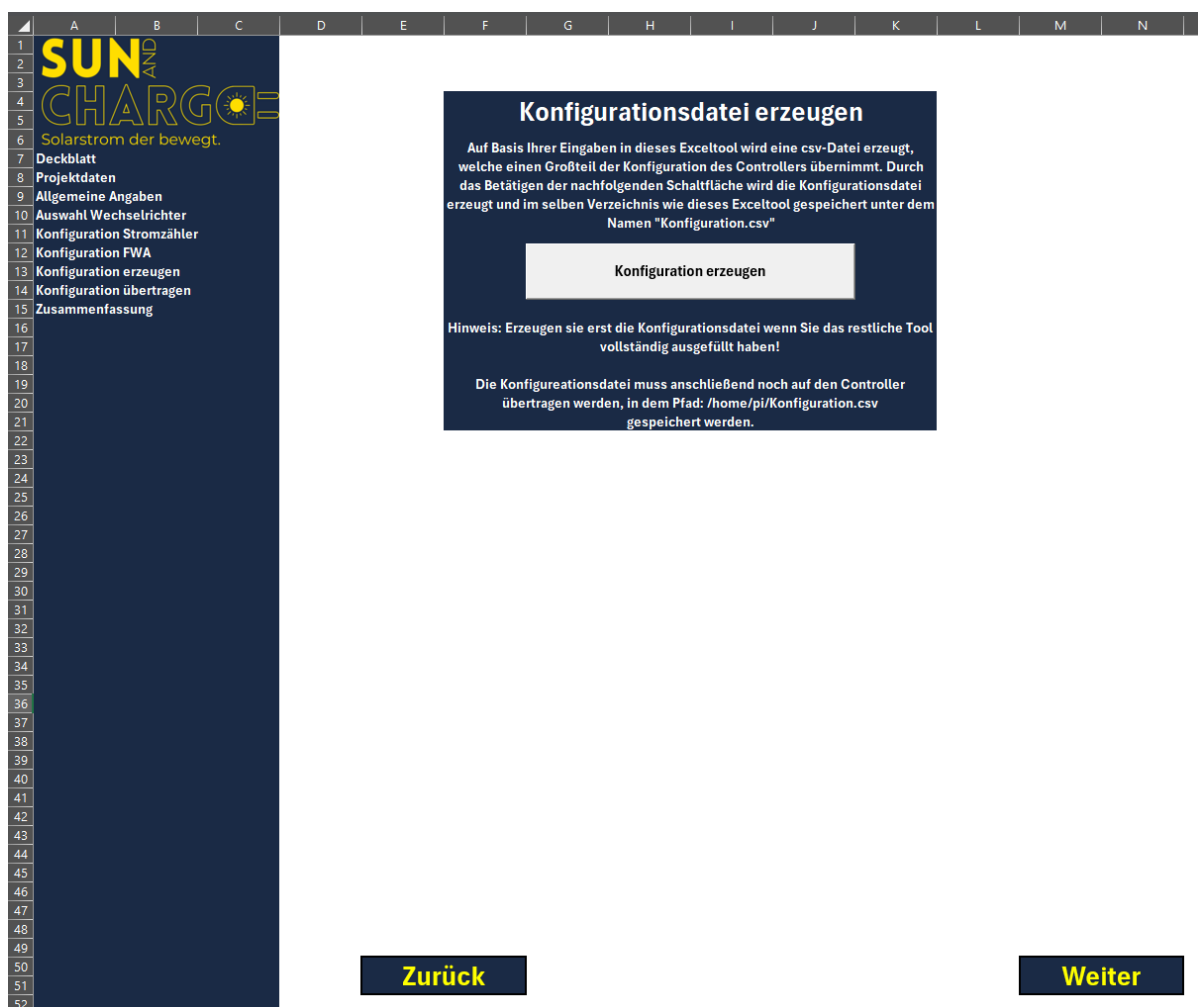
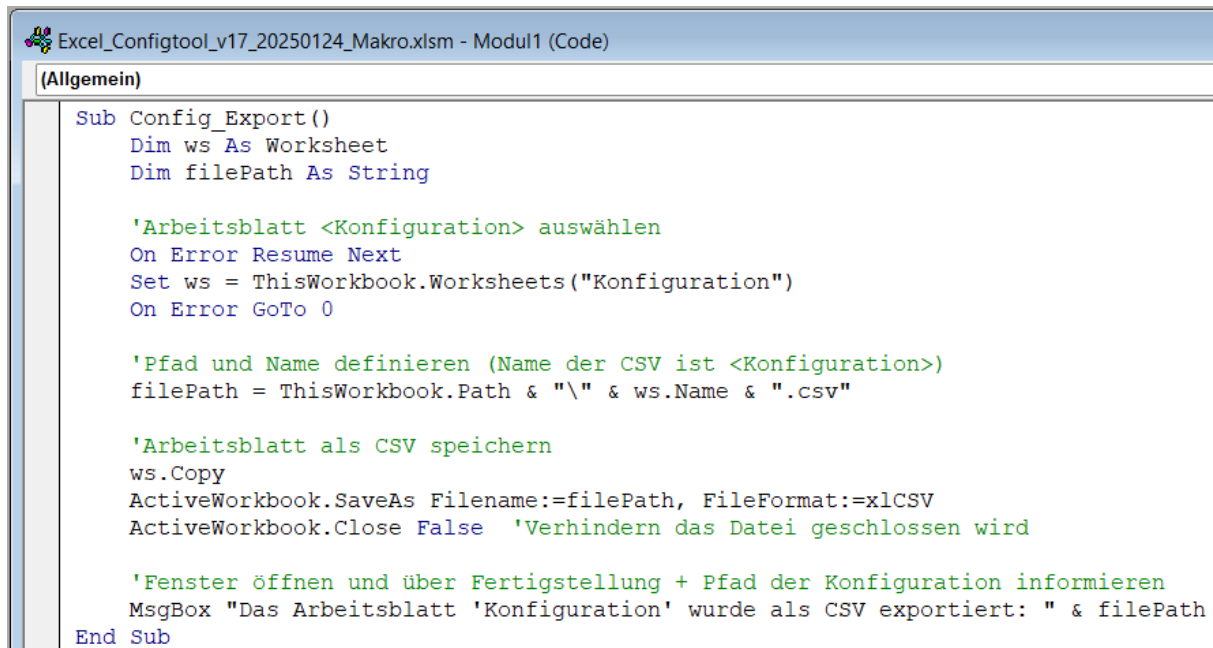


Abbildung 47: Konfigurationstool Tabellenblatt: Konfigurationsdatei erzeugen

Die csv-Konfigurationsdatei wird mit Hilfe eines Visual-Basic-for-Applications-Makros (VBA-Makro) in Excel erzeugt, dies ist in der Abbildung 48 abgebildet. Beim Versenden des

Konfigurationstools können durch Makros immer wieder Probleme mit dem Virenschutz entstehen. Daher wäre es wünschenswert, keine VBA-Makros zu verwenden. Ein Export der Konfiguration ohne die Verwendung von VBA-Makros wäre ebenfalls möglich gewesen, jedoch müsste der Export der Konfiguration manuell vorgenommen werden. Um die Benutzerfreundlichkeit zu erhöhen und Probleme durch falsche Trennzeichen beim manuellen Export zu verhindern, wurde sich dennoch für ein VBA-Makro entschieden.



```
Excel_Configtool_v17_20250124_Makro.xlsm - Modul1 (Code)
(Allgemein)

Sub Config_Export()
    Dim ws As Worksheet
    Dim filePath As String

    'Arbeitsblatt <Konfiguration> auswählen
    On Error Resume Next
    Set ws = ThisWorkbook.Worksheets("Konfiguration")
    On Error GoTo 0

    'Pfad und Name definieren (Name der CSV ist <Konfiguration>)
    filePath = ThisWorkbook.Path & "\" & ws.Name & ".csv"

    'Arbeitsblatt als CSV speichern
    ws.Copy
    ActiveWorkbook.SaveAs Filename:=filePath, FileFormat:=xlCSV
    ActiveWorkbook.Close False 'Verhindern das Datei geschlossen wird

    'Fenster öffnen und über Fertigstellung + Pfad der Konfiguration informieren
    MsgBox "Das Arbeitsblatt 'Konfiguration' wurde als CSV exportiert: " & filePath
End Sub
```

Abbildung 48: VBA-Makro zum Erzeugen der Konfigurationsdatei

Das VBA-Makro wählt das Tabellenblatt `Konfiguration` aus und speichert es im Verzeichnis des Konfigurationstools unter dem Namen „`Konfiguration.csv`“ ab. Anschließend informiert es über ein kleines Fenster über den erfolgreichen Export. Die erzeugte Konfigurationsdatei muss auf dem PV-Controller in dem Verzeichnis `/home/pi/` abgelegt werden und darf nicht umbenannt werden, andernfalls kann der PV-Controller die Konfigurationsdatei nicht finden und kann nicht arbeiten.

In dem Tabellenblatt `Konfiguration` wurde die Konfigurationsdatei vorbereitet. Ein Ausschnitt des Tabellenblatts ist in der Abbildung 49 dargestellt.

	A	B	C	D
1105	Solar control	FWA	FWA current L2 add	0
1106	Solar control	FWA	FWA current L2 qty	0
1107	Solar control	FWA	FWA volatage L1 L3 add	0
1108	Solar control	FWA	FWA volatage L1 L3 qty	0
1109	Solar control	FWA	FWA grid consumption add	0
1110	Solar control	FWA	FWA grid consumption qty	0
1111	Solar control	FWA	FWA reactive power add	0
1112	Solar control	FWA	FWA reactive power qty	0
1113	Solar control	FWA	FWA cos phi add	0
1114	Solar control	FWA	FWA cos phi qty	0
1115	Solar control	FWA	FWA Q P add	0
1116	Solar control	FWA	FWA Q P qty	0
1117	Solar control	FWA	FWA global radiation add	0
1118	Solar control	FWA	FWA global radiation qty	0
1119	Solar control	FWA	FWA emergency stop feedback add	0
1120	Solar control	FWA	FWA emergency stop feedback qty	0
1121	Solar control	PYR	Pyr enable	0
1122	Solar control	PYR	Pyr unit id	0
1123	Solar control	PYR	Pyr read fc	0
1124	Solar control	PYR	Pyr write fc	0
1125	Solar control	PYR	Pyr global radiation add	0
1126	Solar control	PYR	Pyr global radiation qty	0
1127	Solar control	PYR	Pyr global radiation SF	0
1128	Solar control	PYR	Pyr global radiation temp comp add	0
1129	Solar control	PYR	Pyr global radiation temp comp qty	0
1130	Solar control	PYR	Pyr global radiation temp comp SF	0
1131	Solar control	PM1	PM unit id	1
1132	Solar control	PM1	PM enable	1
1133	Solar control	PM1	PM read fc	3
1134	Solar control	PM1	PM write fc	6
1135	Solar control	PM1	PM voltage L1 add	19000
1136	Solar control	PM1	PM voltage L1 qty	2
1137	Solar control	PM1	PM voltage L2 add	19002
1138	Solar control	PM1	PM voltage L2 qty	2


Abbildung 49: Ausschnitt Konfigurationstool Tabellenblatt: Konfiguration

In der ersten Spalte steht für die gesamte Konfiguration des PV-Controllers `Solar control`, dadurch ist die Konfiguration eindeutig diesem Teilbereich des PV-Controllers zuzuordnen. Bei einer Erweiterung des PV-Controllers um beispielsweise Gebäudeautomatisierung kann die benötigte Konfiguration mit z.B. dem Eintrag `Building Automation`

in der ersten Spalte diesem Teil des PV-Controllers zugeordnet werden. In der nächsten Spalte ist das jeweilige Zielgerät für die Konfiguration angegeben. Wie schon in dem Abschnitt 5.1 Konfiguration des PV-Controllers beschrieben, werden die Konfigurationen abhängig von den Eintragungen in den ersten beiden Spalten an den Ziel Flow im PV-Controller weitergeleitet. In der dritten und vierten Spalte sind die Bezeichner mit dem zugehörigen Wert gespeichert. Die Eintragungen in diesen beiden Spalten werden unter Zuhilfenahme der Excel `Übernehmen` Funktion mit den Nutzereingaben gefüllt. Die Modbus-Registerlisten werden ebenfalls abhängig von den ausgewählten Geräten übernommen. In dem Tabellenblatt werden alle Konfigurationen untereinander weggeschrieben, dadurch ist das Tabellenblatt identisch zum Aufbau der csv-Konfigurationsdatei. Eine vollständige Konfigurationsdatei ist im Anhang E zu finden.

6.9 Konfiguration übertragen

Auf dem letzten Tabellenblatt `Konfiguration übertragen` werden zunächst auf der linken Hälfte des Tabellenblattes alle Geräte angezeigt, die für die eine Anpassung auf dem PV-Controller erforderlich sein können. Diese ist in der Abbildung 50 dargestellt. Über die Färbung der Zellen `Aktiv` in Grün oder Rot wird wieder angezeigt, ob ein Gerät konfiguriert ist und die drunter stehende IP-Adresse und Port auf den PV-Controller übertragen werden muss. Zu jedem Gerät wird zusätzlich die Konfiguration angezeigt, in die die Einstellungen übernommen werden müssen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2																
3																
4																
5																
6																
7	Deckblatt															
8	Projektdatei															
9	Allgemeine Angaben															
10	Auswahl Wechselrichter															
11	Konfiguration Stromzähler															
12	Konfiguration FWA															
13	Konfiguration erzeugen															
14	Konfiguration übertragen															
15	Zusammenfassung															
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																
49																
50																
51																
52																
53																
54																
55																
56																
57																
58																
59																
60																
61																
62																
63																
64																
65																
66																
67																
68																
69																
70																

Zu übertragende Konfiguration

Auf Grund von Einschränkungen Seitens Node-RED kann nicht die gesamte Konfiguration eingelesen werden. Die Verbindungseinstellungen der Modbus Flex Getter/Write Nodes lässt sich nicht mit der Konfiguration übertragen und muss händisch vorgenommen werden. Auf Grund dessen, dass diese im Konfigurations-Node gespeichert sind, welcher sich nicht über eine csv-Datei anpassen lassen. In der nachfolgenden Tabelle ist für jedes Gerät über das Feld "Aktiv" in grün gekennzeichnet wenn dieses angeschlossen ist und die Konfiguration übertragen werden muss. Bei einem roten Feld kann dieser Tabelleneintrag ignoriert werden.

Wechselrichter 1 Aktiv: Aktiv IP-Adresse: 10.1.2.100 Port: 502 Konfiguration: Modbus Inv 1	Stromzähler 1 (Bezugszähler) Aktiv: Aktiv IP-Adresse: 10.1.2.110 Port: 502 Konfiguration: Modbus PM 1
Wechselrichter 2 Aktiv: Aktiv IP-Adresse: 10.1.2.101 Port: 502 Konfiguration: Modbus Inv 2	Stromzähler 2 Aktiv: Aktiv IP-Adresse: 10.1.2.111 Port: 502 Konfiguration: Modbus PM 2
Wechselrichter 3 Aktiv: Aktiv IP-Adresse: 10.1.2.102 Port: 502 Konfiguration: Modbus Inv 3	Stromzähler 3 Aktiv: Aktiv IP-Adresse: 10.1.2.112 Port: 502 Konfiguration: Modbus PM 3
Wechselrichter 4 Aktiv: Aktiv IP-Adresse: 10.1.2.103 Port: 502 Konfiguration: Modbus Inv 4	Stromzähler 4 Aktiv: Aktiv IP-Adresse: 10.1.2.113 Port: 502 Konfiguration: Modbus PM 4
Wechselrichter 5 Aktiv: Aktiv IP-Adresse: 10.1.2.103 Port: 502 Konfiguration: Modbus Inv 5	Stromzähler 5 Aktiv: Aktiv IP-Adresse: 10.1.2.114 Port: 502 Konfiguration: Modbus PM 5
Wechselrichter 6 Aktiv: Aktiv IP-Adresse: 10.1.2.104 Port: 502 Konfiguration: Modbus Inv 6	Stromzähler 6 Aktiv: Aktiv IP-Adresse: 10.1.2.115 Port: 502 Konfiguration: Modbus PM 6
FWA Aktiv: Aktiv IP-Adresse: 10.1.2.120 Port: 502 Konfiguration: Modbus FWA	Stromzähler 7 Aktiv: Aktiv IP-Adresse: 10.1.2.116 Port: 502 Konfiguration: Modbus PM 7
Pyranometer Aktiv: Aktiv IP-Adresse: 10.1.2.121 Port: 502 Konfiguration: Modbus Pyr	Stromzähler 8 Aktiv: Aktiv IP-Adresse: 10.1.2.117 Port: 502 Konfiguration: Modbus PM 8
	Stromzähler 9 Aktiv: Aktiv IP-Adresse: 10.1.2.118 Port: 502 Konfiguration: Modbus PM 9

Zurück

Abbildung 50: Linker Teil des Konfigurationstool Tabellenblatt: Konfiguration übertragen

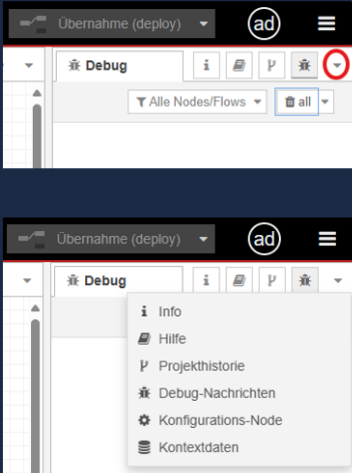
Auf der rechten Seite des Tabellenblatts Konfiguration übertragen ist eine bebilderte Anleitung dargestellt, die dem Nutzer erklärt, wie die Einstellungen auf dem PV-Controller vorgenommen werden. Diese Anleitung ist in der Abbildung 51 abgebildet.

Konfiguration übertragen

- Mit Controller verbinden und Node-RED im Browser aufrufen. Die Node-RED Oberfläche ist über die Adresse <IP_Controller>:41880 zu erreichen (<IP_Controller> durch verwendete IP des Controller ersetzen)

Anmelden in Node-RED mit den bei der Installation von Node-RED festgelegten Anmeldedaten.

Nun muss der Konfigurations-Node geöffnet werden. Dafür muss in der oberen rechten Ecke der Oberfläche auf die eingekreiste Schaltfläche geklickt werden und "Konfigurations-Node" aus der sich öffnenden Liste auswählen.


- Nun wird die Übersicht über alle Konfigurationen des Controllers angezeigt. In dem Abschnitt "Bei allen Flows" werden nun alle Konfigurationen, welche über den Konfigurations-Node vorgenommen werden können, angezeigt.

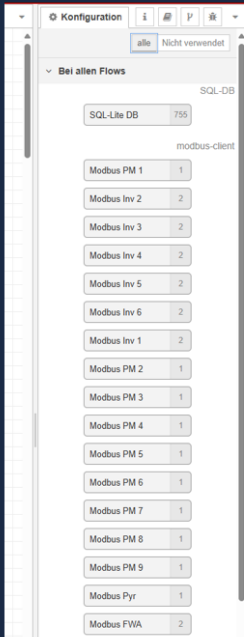

- Zum Übertragen der Konfiguration muss die zum Gerät zugehörige Konfiguration aus der Liste ausgewählt und angepasst werden. Die Zuordnung der Konfiguration an die einzelnen Geräte lassen sich aus der linkstehenden Tabelle aus den Zellen Konfiguration entnehmen.
- Durch klicken auf die Schaltfläche Übernehmen (deploy) Einstellungen übernehmen.

Abbildung 51: Rechter Teil des Konfigurationstool Tabellenblatt: Konfiguration übertragen

Zunächst muss Node-RED im Browser über die Adresse <IP_Controller>:41880 aufgerufen werden. Dabei muss <IP_Controller> durch die IP-Adresse des PV-Controllers ersetzt werden. Anschließend muss sich in Node-RED mit den bei der Installation von Node-RED festgelegten Zugangsdaten angemeldet werden. In Node-RED muss nun der Menüpunkt Konfigurations-Node aufgerufen werden. Danach erscheinen die einzustellenden Konfigurations-Nodes in dem Abschnitt Bei allen Flows unter dem Punkt modbus-client. Nun muss die Konfiguration aus der linken Hälfte des Tabellenblatts in die zugehörigen Konfigurations-Nodes übertragen werden. Nach der vollständigen Übertragung werden die Änderungen durch Betätigen des mit Übernahme (deploy) bezeichneten Knopfes gespeichert und der PV-Controller ist vollständig konfiguriert und kann in Betrieb genommen werden.

7 Erprobung

Die Erprobung des PV-Controllers wird, wie bereits im Abschnitt 1.2 Zielsetzung beschrieben, in einer Testumgebung durchgeführt. Dafür wird zunächst die Testumgebung aufgebaut und die Testfälle für die Erprobung des PV-Controllers werden festgelegt. Anhand der Testfälle wird anschließend die Funktion des PV-Controllers überprüft.

7.1 Testumgebung

Das Testen des PV-Controllers stellt eine große Herausforderung dar, denn eine Erprobung ist am Livesystem nicht möglich, da eine korrekte Funktion nicht sicherzustellen ist. Ein etwaiger Ausfall könnte bei sich in Betrieb befindlichen Anlagen wirtschaftliche Folgen nach sich ziehen. Um dieses Problem zu lösen, wird zur Erprobung und Sicherstellung der korrekten Funktion des PV-Controllers eine Testumgebung aufgebaut. Zusätzlich hat die Testumgebung den Vorteil, dass die Entwicklung des PV-Controllers und diese Arbeit unabhängig von dem begleiteten Projekt ist. In der Testumgebung werden die an den PV-Controller angeschlossenen Geräte von einem Loytec LINX-120 (LINX-120) simuliert. Dies hat den Vorteil, dass zum Testen des PV-Controllers nicht die im Feld verwendeten Geräte benötigt werden und auch Betriebssituationen, welche im Regelbetrieb nicht auftreten sollten, getestet werden können. So beispielsweise die Situation, dass die PV-Anlage durch die FWA gedrosselt wird. Diese Betriebssituation wird im Regelbetrieb nicht eintreten, da bei dem begleiteten Projekt der Netzbezug so groß ist, dass eine Einspeisung nur dann auftritt, wenn große Teile der Produktion abgeschaltet sind. Zusätzlich müsste in diesem Zeitraum zur Überprüfung der Funktion ein Steuersignal des Netzbetreibers zur Drosselung der Anlage übermittelt werden.

Der schematische Aufbau der Testumgebung ist in der nachfolgenden Abbildung 52 dargestellt.

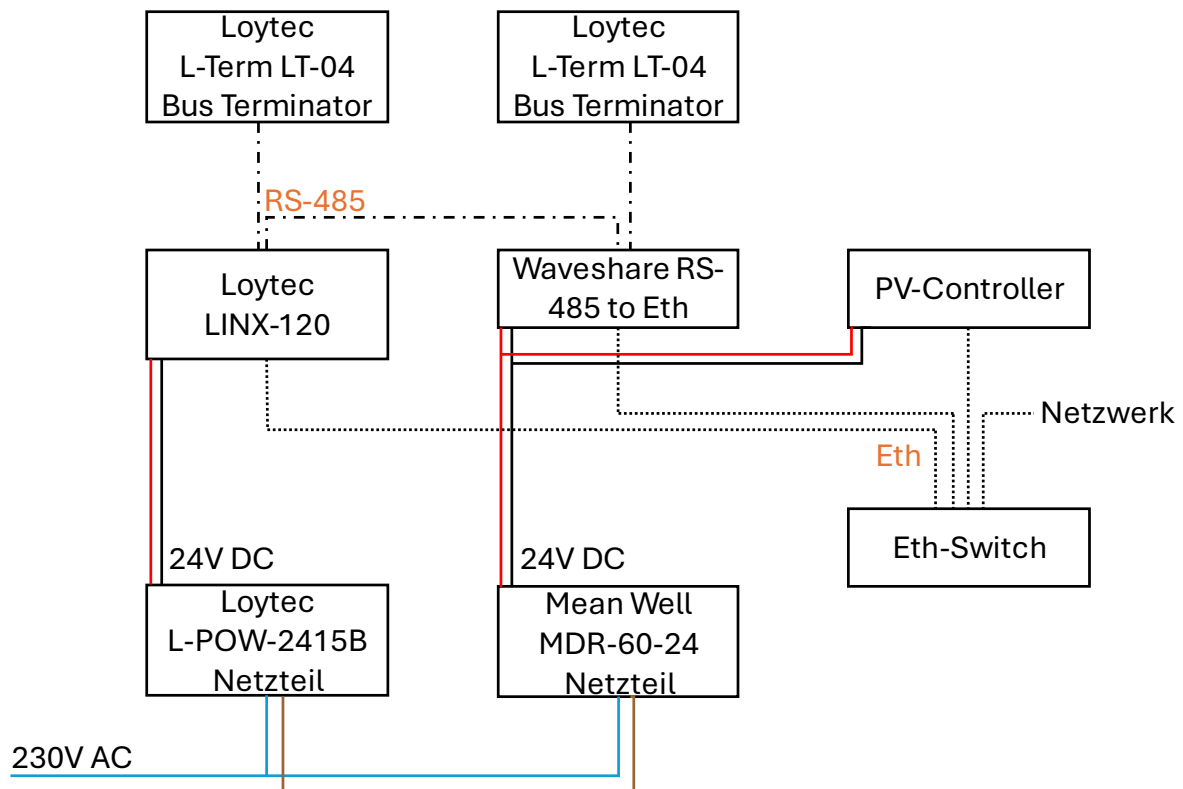


Abbildung 52: Schematische Darstellung des Testaufbaus

Der Testaufbau wird in einem Schaltkasten realisiert. Für die Stromversorgung werden Hutschienen Netzteile mit 24 V DC verwendet. Über Ethernet werden der LINX-120 und der Waveshare RS-485 zu TCP/IP Umsetzer über einen Switch an den PV-Controller angeschlossen. Zwischen den LINX-120 und der Waveshare RS-485 zu TCP/IP Umsetzer wird ein RS-485 Bus angeschlossen. Dieser ist durch Busabschlüssen (Loytec L-Term LT-04) terminiert. Somit ist es möglich, die Modbus TCP und die umgesetzte Modbus RTU Verbindung zu erproben.

Nun müssen die angeschlossenen Geräte konfiguriert werden. Zur Konfiguration des LINX-120 wird das LINX Configurator Programm von Loytec verwendet. Dies kann von der Loytec-Webseite heruntergeladen werden [52]. Unter Zuhilfenahme dieses Programms lässt sich die Modbus TCP und Modbus RTU Verbindung konfigurieren. Auf dem LINX-120 werden die jeweiligen Modbusregister der simulierten Geräte konfiguriert.

Die Modbus TCP Register können über die IP-Adresse 10.1.2.250 und den Port 502 erreicht werden. Zur Modbus-RTU-Kommunikation wird der RS-485 zu TCP/IP Umsetzer benötigt. Bei der Konfiguration ist darauf zu achten, dass die Baudrate, die Parität, die Anzahl an Datenbits und die Anzahl der Stop-Bits identisch zu den Einstellungen auf dem LINX-120 sind, um Fehler bei der Kommunikation zu vermeiden. Beim Einsatz im Feld müssen diese Einstellungen an das jeweilige angeschlossene Gerät angepasst werden. Der RS-485 zu TCP/IP Umsetzer ist unter der IP-Adresse 10.1.2.249 und dem Port 502 zu erreichen. Bei der Kommunikation über den RS-485 zu TCP/IP Umsetzer ist die zuvor am LINX-120 eingestellte `Unit-ID` 1 zu verwenden.

Abschließend wird die Modbus-TCP-Verbindung zum LINX-120 und zum RS-485 zu TCP/IP Umsetzer getestet. Dafür wird das Programm `qmodMaster` verwendet [53]. Mit diesem Programm lassen sich Modbus-TCP-Verbindungen testen. Es wird sich jeweils mit dem Linx-120 und dem RS-485 zu TCP/IP Umsetzer verbunden und es wird überprüft, ob die zuvor im LINX-Configurator eingestellten Modbus Register zu erreichen sind. In Abbildung 53 ist der Test der Modbus Kommunikation mit dem RS-485 zu TCP/IP Umsetzer für das Register mit der Adresse 30775 dargestellt.

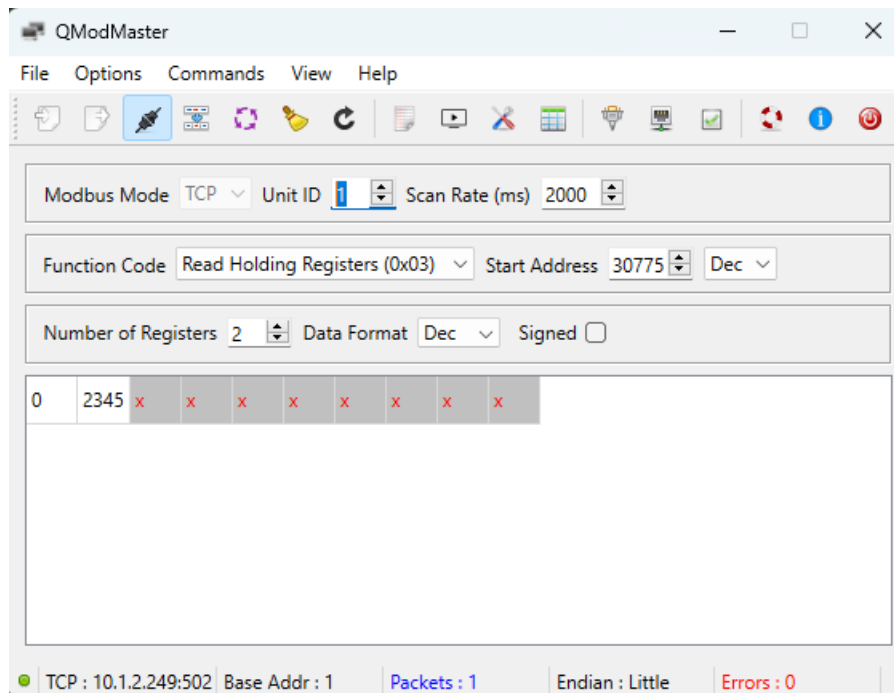


Abbildung 53: Test der Modbus Kommunikation des RS-485 zu TCP/IP Umsetzer für das Register mit der Adresse 30775

Es ist darauf zu achten, dass bei der Verbindung mit dem LINX-120 die Register der konfigurierten Modbus TCP Verbindung angezeigt werden. Bei der Verbindung mit dem RS-485 zu TCP/IP Umsetzer müssen die konfigurierten Modbus RTU Register angezeigt werden. Im weiteren Verlauf der Erprobung wird der RS-485 zu TCP/IP Umsetzer nicht weiterverwendet, da die Funktion bereits getestet wurde.

7.2 Testfälle

Für die Sicherstellung der korrekten Funktion des PV-Controllers werden Testfälle definiert, anhand deren der PV-Controller auf seine korrekte Funktion getestet werden kann. Es wurden folgende Testfälle definiert:

Testfall 1: Einlesen der Konfiguration

Ein Großteil der Konfiguration wird über eine csv-Datei eingelesen. Das korrekte Einlesen dieser Datei muss überprüft werden. Folgende Funktionen müssen dafür getestet werden:

- Konfigurationsdatei hat den richtigen Aufbau.
- Die Konfigurationsdatei wurde in dem Dateipfad `/home/pi/` des PV-Controllers gespeichert.
- Der PV-Controller liest die Konfigurationsdatei ein und speichert die Konfiguration vollständig in den richtigen Flows ab.

Testfall 2: FWA Nennleistungssteuerung

Die richtige Umsetzung der Nennleistungssteuerungsbefehle der FWA muss überprüft werden. Diese wird von der FWA in % (bezogen auf die installierte Nennleistung) angegeben.

Folgende Funktionen müssen dafür getestet werden:

- Steuersignal wird von der FWA empfangen.
- Steuersignal wird korrekt ausgewertet und es wird eine Wirkleistungsbegrenzung an den WR übertragen.
- Umsetzung des Steuerbefehls zurückmelden an FWA.
- Umsetzung innerhalb von maximal 60 Sekunden.

Testfall 3: FWA Blindleistungssteuerung

Die richtige Umsetzung der Blindleistungssteuerungsbefehle der FWA muss überprüft werden. Dabei kann vom Versorgungsnetzbetreiber zwischen einem Kennlinienbetrieb und der Steuerung durch die NLS umgeschaltet werden. Der Blindleistungssteuerbefehl ist in % (bezogen auf die installierte Leistung) angegeben.

Folgende Funktionen müssen dafür getestet werden:

- Die Steuersignale Q(P) Steuerung und Blindleistungssteuerung werden von der FWA abgefragt.
- Der richtige Steuermodus wird abhängig vom Steuersignal Q(P) Steuerung gewählt.
- Rückmeldung an FWA, welcher Steuermodus verwendet wird.
- Bei Blindleistungssteuerung durch NLS. Überprüfen, ob das Steuersignal korrekt ausgewertet wird und die Blindleistungsbegrenzung an den Wechselrichter übertragen wird.
- Umsetzung innerhalb von maximal 60 Sekunden.

Testfall 4: FWA Messwerte Bezugszähler

Zur korrekten Funktion der FWA werden Messwerte vom Bezugszähler benötigt. Es werden die Messwerte Strom L2 (IL2), Spannung L1L2 (UL1_L2), Wirkleistung (P), Blindleistung Q (Q), Phasenverschiebung (cos phi) und Q/P_{inst} in % von der FWA benötigt.

Folgende Funktionen müssen dafür getestet werden:

- Messwerte werden vom Stromzähler abgefragt.
- Die Phasenverschiebung (cos phi) und Q/P_{inst} in % werden korrekt berechnet.
- Die Messwerte werden an die FWA übertragen.

Testfall 5: FWA Messwerte Pyranometer

Von der FWA wird die Globalstrahlung benötigt. Diese wird von einem Pyranometer abgefragt.

Folgende Funktionen müssen dafür getestet werden:

- Die Globalstrahlung wird vom Pyranometer abgefragt.
- Die Globalstrahlung wird an die FWA übertragen.

Testfall 6: FWA Not-Aus

Der Netzbetreiber hat über die FWA die Möglichkeit über einen Not-Aus Steuerbefehl die Erzeugungsanlagen vom Netz zu trennen. Dieses Signal wird bisher nicht von dem PV-Controller umgesetzt es wird nur abgerufen und auf dem PV-Controller für eine spätere Verwendung

Folgende Funktionen müssen dafür getestet werden:

- Abfrage des Not-Aus Steuerbefehls.
- Speicher des Not-Aus Steuerbefehls in einer globalen Variable auf dem PV-Controller.

Testfall 7: Begrenzung der Einspeiseleistung

Die maximale Einspeiseleistung wird mit dem Netzbetreiber vereinbart und darf nur kurzzeitig überschritten werden. Bei einem zu langen Überschreiten werden die Erzeugungsanlagen durch die $P_{AV,E}$ -Überwachung vom Netz getrennt. Die im Bau verwendete $P_{AV,E}$ -Überwachung besitzt keine geeignete Kommunikationsschnittstelle, um eine nötige Einspeiseleistungsreduzierung vorzunehmen. Deshalb soll diese von dem PV-Controller übernommen werden. Um ein Trennen vom Netz zu verhindern, werden die Wechselrichter bereits vor Erreichen der Grenze gedrosselt. Voraussetzung dafür ist, dass von der FWA keine andere Wirkleistungsvorgabe gemacht wird.

Folgende Funktionen müssen dafür überprüft werden:

- Messwerte werden vom Bezugszähler abgefragt.
- Korrekte Auswertung, ob eine Leistungsreduzierung der Wechselrichter vorgenommen werden muss.
- Reduzierung der Leistung der Wechselrichter.

Testfall 8: Reduzierung von Lastspitzen

Der maximale Netzbezug wird mit dem Netzbetreiber vereinbart und für jede Lastspitze (15 Minuten Mittelwert des Netzbezuges) über diesem Netzbezug werden Strafzahlungen fällig. Zur Reduzierung der Lastspitzen wird vom PV-Controller das System innerhalb dessen Systemgrenzen betrachtet und für den Fall, dass eine Leistungsreduzierung aus Sicht des PV-Controllers als notwendig erachtet wird, gibt der PV-Controller einen Hinweis zur Leistungsreduzierung aus.

Folgende Funktionen müssen dafür getestet werden:

- Der Netzbezug muss vom Bezugszähler abgefragt werden.
- Korrekte Auswertung, ob eine Leistungsreduzierung aus Sicht des PV-Controller notwendig ist.
- Speichern des Ergebnisses in einer globalen Variable (0: keine Leistungsreduzierung, 1: Leistungsreduzierung) auf dem PV-Controller.

Testfall 9: Funktion der JEVIs Datenbank

Die abgefragten Register der verbundenen Geräte werden für das Energiemonitoring in JEVIs in einer Datenbank abgespeichert. Es muss überprüft werden, ob die abgefragten Register mit in die Datenbank für JEVIs aufgenommen werden.

Folgende Funktionen müssen dafür getestet werden:

- Alle Register der Geräte werden abgefragt.
- Die Intervall Nodes, welche die Häufigkeit der Messwerterfassung für JEVIs festlegen sind mit den richtigen Intervallen konfiguriert (Einen Wert alle viertel Stunde (xx:00, xx:15, xx:30, xx:45)).
- Die Dauer des Verbleibs im Ringspeicher wurde richtig aus der Konfiguration übernommen.
- Die abgefragten Register werden vollständig in den Datenbank Node übertragen.

7.3 Durchführung der Erprobung

Im Folgenden werden die Testfälle nacheinander abgearbeitet und die korrekte Funktion des PV-Controllers wird überprüft. Für die Erprobung wurde die zyklische Abfrage der verbundenen Geräte deaktiviert und die Abfragen der Geräte werden manuell gestartet. Dadurch ist es möglich, die einzelnen Funktionen zu testen, ohne das Register bei der Abfrage mit neueren Daten überschrieben werden. Somit lässt sich die Funktion des Controllers nachvollziehen und überprüfen.

7.3.1 Auswertung Testfall 1: Einlesen der Konfiguration

Entsprechend des Testfalls 1 aus Abschnitt 7.2 muss überprüft werden, ob die Konfiguration richtig eingelesen wird. Es wird zunächst überprüft, ob die Konfigurationsdatei den richtigen Aufbau hat. Dafür wird eine Konfiguration mit Hilfe des Excel-Konfigurations-tools erzeugt. Anschließend wird die Konfigurationsdatei mit den Angaben in dem Konfigurationstool abgeglichen. Dabei wird überprüft, ob die richtigen Informationen in die Konfigurationsdatei übernommen wurden. Es wurde festgestellt, dass die richtigen Daten übernommen werden.

Anschließend wird die Konfigurationsdatei im Verzeichnis `/home/pi/` abgelegt und der PV-Controller bzw. Node-RED wird gestartet. Beim Starten des PV-Controllers liest dieser

selbständig die Konfigurationsdatei ein. In Node-RED wird anschließend überprüft, ob die Konfiguration vollständig eingelesen wurde. Dabei wird in jedem `Flow` die Konfiguration betrachtet und geprüft, ob diese vollständig ist. Bei der Überprüfung des Einlesens der Konfiguration wurde festgestellt, dass die Konfigurationsdatei korrekt eingelesen wird.

7.3.2 Auswertung Testfall 2: FWA Nennleistungssteuerung

Entsprechend des Testfalls 2 aus Abschnitt 7.2 wird zur Überprüfung der korrekten Umsetzung des Nennleistungssteuerungsbefehl der FWA, auf dem LINX-120 eine Nennleistungsvorgabe der FWA von 60% simuliert. In der Abbildung 54 sind die für den Test relevanten Register zu Beginn der Überprüfung abgebildet. Der Nennleistungssteuerungsbefehl (`FWA power control target`) der FWA ist auf 60 % gesetzt.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	60
FWA power control feedback	value	analog	normal	0
Inv WMaxLimPct	value	analog	normal	0
Inv WMaxLimPct RvrtTms	value	analog	normal	0
Inv WMaxLimPct Ena	value	analog	normal	0

Abbildung 54: Nennleistungsvorgabe 60% durch FWA

Anschließend wird auf dem PV-Controller die Abfrage der FWA gestartet und die Antwort abgewartet. In der Abbildung 55 sind die Register nach der Antwort des PV-Controllers abgebildet. Der Wechselrichter bekommt in dem Register `Inv WMaxLimPct` die Nennleistungsvorgabe von 60% übergeben. In dem Register `Inv WMaxLimPct RvrtTms` wird der Timer für die Dauer der Nennleistungsvorgabe auf 180 Sekunden gesetzt, um sicherzustellen, dass die Wirkleistungsvorgabe bis zum erneuten Setzen aktiv bleibt. Mit dem Setzen des Registers `Inv WMaxLimPct Ena` auf 1 wird die Nennleistungsvorgabe auf dem Wechselrichter aktiviert. Die Rückmeldung an die FWA an das Register `FWA power control feedback` wird ebenfalls korrekt ausgeführt.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	60
FWA power control feedback	value	analog	normal	60
Inv WMaxLimPct	value	analog	normal	60
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 55: Antwort nach Nennleistungsvorgabe 60% durch FWA

Anschließend wird überprüft, ob die Nennleistungsvorgabe des Wechselrichters auch korrekt wieder deaktiviert wird, wenn die FWA eine Nennleistungsvorgabe von 100% vorgibt. Dies ist in der Abbildung 56 dargestellt.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	100
FWA power control feedback	value	analog	normal	60
Inv WMaxLimPct	value	analog	normal	60
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 56: Nennleistungsvorgabe 100% durch FWA

Die Antwort des PV-Controllers ist in der Abbildung 57 dargestellt. Durch korrektes Rücksetzen des Registers `Inv WMaxLimPct Ena` auf 0, wird die Nennleistungsvorgabe auf dem Wechselrichter deaktiviert. Die Umsetzung der neuen Nennleistungsvorgabe wird korrekt an das Register `FWA power control feedback` der FWA rückgemeldet.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	100
FWA power control feedback	value	analog	normal	100
Inv WMaxLimPct	value	analog	normal	60
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	0

Abbildung 57: Antwort nach Nennleistungsvorgabe 100% durch FWA

Auf Grund der zyklischen Abfrage der FWA in 10-Sekunden-Intervallen wird der Nennleistungssteuerungsbefehl sicher innerhalb von den geforderten 60 Sekunden umgesetzt.

7.3.3 Auswertung Testfall 3: FWA Blindleistungssteuerung

Entsprechend des Testfalls 3 aus Abschnitt 7.2 wird die korrekte Umsetzung der Blindleistungssteuerungsbefehle der FWA überprüft. Dafür wird der Steuermodus der Blindleistung (`FWA reactive power control mode`) von dem Kennlinienbetrieb, mit der im Wechselrichter hinterlegten Kennlinie, zur Blindleistungsvorgabe durch die NLS umgeschaltet. Zusätzlich wird eine Blindleistungsvorgabe (`FWA reactive power control target`) von 20 % (bezogen auf die installierte Leistung) von der FWA vorgegeben. In der Abbildung 58 sind die Blindleistungssteuerungsbefehle der FWA abgebildet.

Name	Richt.	Typ	Status	Wert
<code>FWA reactive power control mode</code>	value	analog	normal	1
<code>FWA reactive power control target</code>	value	analog	normal	20
<code>FWA reactive power control mode feedback</code>	value	analog	normal	0
<code>Inv VArWMaxPct</code>	value	analog	normal	0
<code>Inv VArWMaxPct1 RvrtTms</code>	value	analog	normal	0
<code>Inv VArWMaxPct1 Ena</code>	value	analog	normal	0

Abbildung 58: Umschaltung des Steuermodus zur Blindleistungsvorgabe durch NLS und Blindleistungsvorgabe von 20 %

Anschließend wird die Abfrage der FWA gestartet und die Antwort des PV-Controllers abgewartet. Das Ergebnis ist in der Abbildung 59 dargestellt. Dem Wechselrichter wird in dem Register `Inv VArWMaxPct` die Blindleistungsvorgabe von 20 % übergeben. Wie auch bei der Nennleistungssteuerung wird ein Timer auf 180 Sekunden gesetzt, um sicherzustellen, dass die Blindleistungsvorgabe eingehalten wird. Diese Zeit wird in das Register `Inv VArWMaxPct RvrtTms` geschrieben. Durch Setzen des Registers `Inv VArWMaxPct Ena` wird die Blindleistungsvorgabe aktiviert. Der FWA wird die Änderung des Steuermodus der Blindleistung durch Setzen des Registers `FWA reactive power control mode feedback` zurückgemeldet.

Name	Richt.	Typ	Status	Wert
FWA reactive power control mode	value	analog	normal	1
FWA reactive power control target	value	analog	normal	20
FWA reactive power control mode feedback	value	analog	normal	1
Inv VArWMaxPct	value	analog	normal	20
Inv VArWMaxPct1 RvrtTms	value	analog	normal	180
Inv VArWMaxPct1 Ena	value	analog	normal	1

Abbildung 59: Antwort Umschaltung des Steuermodus zur Blindleistungsvorgabe durch NLS und Blindleistungsvorgabe von 20 %

Im Anschluss wird überprüft, ob sich der Steuermodus der Blindleistung wieder auf den Kennlinienbetrieb umschalten lässt. Dafür wurde das Register `FWA reactive power control mode` der FWA wieder rückgesetzt, dies ist in der Abbildung 60 dargestellt.

Name	Richt.	Typ	Status	Wert
FWA reactive power control mode	value	analog	normal	0
FWA reactive power control target	value	analog	normal	20
FWA reactive power control mode feedback	value	analog	normal	1
Inv VArWMaxPct	value	analog	normal	20
Inv VArWMaxPct1 RvrtTms	value	analog	normal	180
Inv VArWMaxPct1 Ena	value	analog	normal	1

Abbildung 60: Umschaltung des Steuermodus zurück in den Kennlinienbetrieb

Nach der Abfrage der FWA wird der Steuermodus der Blindleistung vom PV-Controller zurück zum Kennlinienbetrieb geändert. Die Antwort des PV-Controllers ist in der Abbildung 61 abgebildet. Durch Rücksetzen des Registers `Inv VArWMaxPct Ena` wird die externe Blindleistungssteuerung des Wechselrichters deaktiviert und er arbeitet weiter nach der hinterlegten Blindleistungskennlinie. Durch das Rücksetzen des Registers `FWA reactive power control mode feedback` wird die Umschaltung zurück auf den Kennlinienbetrieb der FWA rückgemeldet.

Name	Richt.	Typ	Status	Wert
FWA reactive power control mode	value	analog	normal	0
FWA reactive power control target	value	analog	normal	20
FWA reactive power control mode feedback	value	analog	normal	0
Inv VArWMaxPct	value	analog	normal	20
Inv VArWMaxPct1 RvrtTms	value	analog	normal	180
Inv VArWMaxPct1 Ena	value	analog	normal	0

Abbildung 61: Antwort Umschaltung des Steuermodus zurück in den Kennlinienbetrieb

Die Blindleistungssteuerungsbefehle der FWA werden korrekt umgesetzt und wie auch bei der FWA Nennleistungssteuerung werden die Steuerbefehle innerhalb von 60 Sekunden umgesetzt. Dies geschieht auf Grund der zyklischen Abfrage der FWA alle 10 Sekunden.

7.3.4 Auswertung Testfall 4: FWA Messwerte Bezugzähler

Bei der Durchführung des in Abschnitt 7.2 beschriebenen Testfalls 4 ist aufgefallen, dass die Modbus-Register des verwendeten Stromzählers nicht korrekt ausgelesen werden. Grund dafür ist, dass der Stromzähler die Daten als Float-Werte mit einer Länge von 32-Bit speichert. Beim Abrufen dieser über Modbus müssen zwei Register gelesen werden. Dies wird auf dem PV-Controller zwar korrekt getan, jedoch muss die Antwort des `Modbus-Flex-Getter-Nodes` erst durch einen zusätzlichen Buffer Parser in die richtige Form gebracht werden. Um dieses Problem zu lösen, wurde eine geeignete Palette ausgewählt, mit deren Hilfe die Register korrekt eingelesen werden können. Es wurde die Palette `node-red-contrib-buffer-parser` ausgewählt [54]. Die Installation dieser Palette ist fehlgeschlagen, wodurch sich diese aktuell nicht verwenden lässt. Das Problem ist voraussichtlich mit einer zukünftigen Aktualisierung der Node-RED Umgebung, auf Grund von gelösten Software-Abhängigkeiten im Paketmanager NPM, gelöst. Alternativ besteht die Möglichkeit, Paletten direkt über die Konsole manuell zu installieren. Dies wurde jedoch nicht getan, weil auf diesem Weg keine standardisierte Produktlösung geschaffen werden kann, bei der sich die benötigten Paletten automatisch beim Importieren

des PV-Controllers in Node-RED installieren. Alternativ könnte ein eigener Buffer Parser in Zukunft entwickelt und der Node-RED Community zur Verfügung gestellt werden.

Beim Senden der Messwerte an die FWA besteht das gleiche Problem. Es wird für die Übertragung von Werten, welche über mehrere Register gespeichert sind, ein Buffer Parser benötigt, um die zu übertragenden Daten korrekt vorzubereiten. Um dennoch die Berechnung der Phasenverschiebung und Q/P_{inst} in % zu überprüfen, wird den Function Nodes in dem diese berechnet werden über Inject Nodes Messwerte des Stromzählers vorgegeben. In der Abbildung 62 ist dies dargestellt. Dabei wird eine Wirkleistung von 100 kW, eine Scheinleistung von 120 kVA und eine Blindleistung von 20 kvar vorgegeben. Die Phasenverschiebung und Q/P_{inst} in % werden korrekt berechnet. Das Ergebnis der Berechnung wird an den debug Nodes ausgegeben.

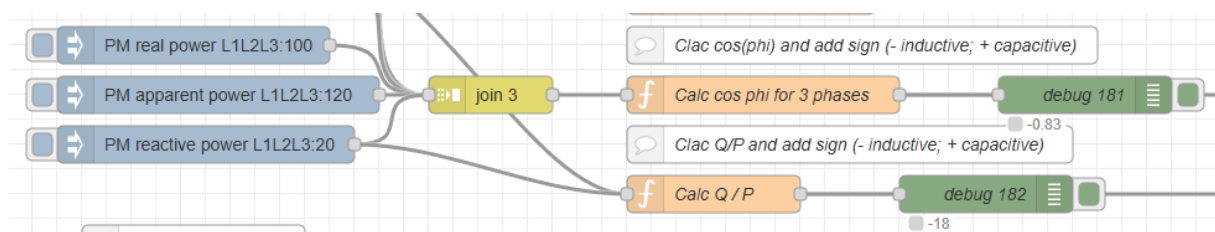


Abbildung 62: Test der Function Nodes zur Berechnung der Phasenverschiebung $\cos(\phi)$ und Q/P installiert in %

Auf Grund der Anforderungen der WEMAG muss auf die Vorzeichen geachtet werden. So wird bei induktiver Blindleistung ein negatives Vorzeichen an die berechneten Werte angehängt.

7.3.5 Auswertung Testfall 5: FWA Messwerte Pyranometer

Zur Erprobung der Übertragung der Globalstrahlung an die FWA wird entsprechend des Testfalls 5 aus Abschnitt 7.2, von dem simulierten Pyranometer eine Globalstrahlung (`PYR global radiation`) von 1230 W/m^2 vorgegeben. Auf dem PV-Controller muss diese mit dem Faktor 10 skaliert werden. Der Skalierungsfaktor wird in der Konfiguration mit übergeben. In der Abbildung 63 ist die Vorgabe der Globalstrahlung dargestellt.

Name	Richt.	Typ	Status	Wert
PYR global radiation	value	analog	normal	123
FWA global radiation	value	analog	normal	0

Abbildung 63: Globalstrahlung des Pyranometers vorgeben

In der Abbildung 64 ist die Antwort des PV-Controllers dargestellt. An die FWA wird in das Register `FWA global radiation` die korrekt skalierte Globalstrahlung übertragen.

Name	Richt.	Typ	Status	Wert
PYR global radiation	value	analog	normal	123
FWA global radiation	value	analog	normal	1230

Abbildung 64: Antwort auf Vorgabe der Globalstrahlung des Pyranometers

7.3.6 Auswertung Testfall 6: FWA Not-Aus

Entsprechend des Testfalls 6 aus Abschnitt 7.2 wird überprüft, ob das Not-Aus-Signal korrekt von der FWA abgefragt wird und auf dem PV-Controller zur weiteren Verwendung gespeichert wird. Zur Überprüfung der Funktion wird von der FWA als Not-Aus-Signal (`FWA emergency stop`) mit dem Wert 1 vorgegeben und die Antwort des PV-Controllers wird abgewartet.

Name	Richt.	Typ	Status	Wert
FWA emergency stop	value	analog	normal	1

Abbildung 65: FWA Not-Aus-Signal vorgeben

Von dem PV-Controller wird das Not-Aus-Signal korrekt ausgelesen und wie in Abbildung 66 zu erkennen in der globalen Variable `FWA.emergency_stop` gespeichert.

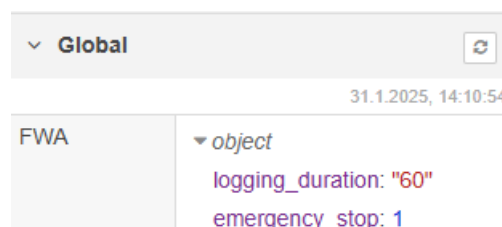


Abbildung 66: Gespeichertes Not-Aus-Signal auf dem PV-Controller

7.3.7 Auswertung Testfall 7: Begrenzung der Einspeiseleistung

Auf Grund des, in Auswertung Testfall 4: FWA Messwerte Bezugszähler beschriebenen, Problems mit dem Einlesen von 32-Bit-Datentypen mittels Modbus, kann in diesem Testfall der Netzbezug (PM real power L1L2L3) nicht vom Bezugszähler abgefragt werden. Um diese Funktion entsprechend des Testfalls 7 aus Abschnitt 7.2 trotzdem testen zu können, wird ein Netzbezug von -250 kW mit einem Inject Node auf dem PV-Controller vorgegeben. Dies entspricht einer Einspeiseleistung von 250 kW. Der PV-Controller benötigt ebenfalls die in der Konfiguration übergebenen maximale Einspeiseleistung in das Versorgungsnetz, die in der Konfiguration auf 270 kW festgelegt wurde.

Die PV-Anlagen werden in 20 %-Schritten gedrosselt, sobald die Einspeiseleistung 20 kW oder weniger von der maximalen Einspeiseleistung in das Versorgungsnetz entfernt ist. Mit dem eingestellten Netzbezug von -250 kW ist dies der Fall und eine Leistungsreduzierung muss vorgenommen werden. Voraussetzung dafür, dass eine Leistungsanpassung vorgenommen werden kann, ist eine Wirkleistungsvorgabe von 100 % der FWA. Denn die Steuerbefehle der FWA haben Vorrang vor einer Leistungsanpassung zur Unterstützung der $P_{AV,E}$ -Überwachung. In der Abbildung 67 ist zu erkennen, dass dem PV-Controller von der FWA eine Wirkleistungsvorgabe (FWA power control target) von 100% vorgegeben wird.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	100
FWA power control feedback	value	analog	normal	0
Inv WMaxLimPct	value	analog	normal	100
Inv WMaxLimPct RvrtTms	value	analog	normal	0
Inv WMaxLimPct Ena	value	analog	normal	0

Abbildung 67: Begrenzung der Einspeiseleistung bei 100 % Vorgabe der FWA

Von dem PV-Controller werden die Wechselrichter auf 80 % gedrosselt und dem Wechselrichter wird die aktuelle Leistungsreduzierung rückgemeldet. Zur Leistungsreduzierung werden dieselben Register des Wechselrichters, wie auch in der Auswertung Testfall 2: FWA Nennleistungssteuerung, beschrieben. Dies ist in der Abbildung 68 zu erkennen.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	100
FWA power control feedback	value	analog	normal	80
Inv WMaxLimPct	value	analog	normal	80
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 68: Antwort Begrenzung der Einspeiseleitung

In dem Fall, dass die Einspeiseleistung sich nicht verringert hat – etwa, wenn zeitgleich Verbraucher abgeschaltet wurden – muss die Leistung weiter reduziert werden. Dies geschieht bei der nächsten Abfrage der FWA, wenn der PV Control Flow erneut ausgeführt wird, wie in der Abbildung 69 dargestellt. Es ist zu erkennen, dass die Wechselrichter um weitere 20 % gedrosselt wurden.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	100
FWA power control feedback	value	analog	normal	60
Inv WMaxLimPct	value	analog	normal	60
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 69: Antwort Begrenzung der Einspeiseleitung bei unverändert zu hoher Einspeiseleistung

Wenn die Einspeiseleistung sich verringert hat und wieder mehr als 20 kW von der maximalen Einspeiseleistung entfernt ist, wird die Leistung der Wechselrichter in 20 % Schritten wieder erhöht. Um dies zu testen, wird eine Wirkleistung von -249 kW am NAP vorgegeben. In der Abbildung 70 ist die Antwort des PV-Controller abgebildet. Es ist zu erkennen, dass die Leistung der Wechselrichter wieder um 20 % erhöht wurde.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	100
FWA power control feedback	value	analog	normal	80
Inv WMaxLimPct	value	analog	normal	80
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 70: Antwort Begrenzung der Einspeiseleitung bei Einspeiseleistung unterhalb der Grenze zur Leistungsreduzierung

Wenn von der FWA eine Wirkleistungsvorgabe gemacht wird, wird diese unverzüglich nach dem Empfang wieder umgesetzt. Im nächsten Schritt wird von der FWA eine Wirkleistungsvorgabe von 60 % gemacht. Dies ist in der Abbildung 71 dargestellt.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	60
FWA power control feedback	value	analog	normal	80
Inv WMaxLimPct	value	analog	normal	80
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 71: Begrenzung der Einspeiseleitung bei Wirkleistungsvorgabe durch FWA

In der Abbildung 72 ist zu erkennen, dass die Wirkleistungsvorgabe der FWA Vorrang hat und unabhängig von der ggf. benötigten Leistungsreduzierung zur Begrenzung der Einspeiseleitung umgesetzt wird.

Name	Richt.	Typ	Status	Wert
FWA power control target	value	analog	normal	60
FWA power control feedback	value	analog	normal	60
Inv WMaxLimPct	value	analog	normal	60
Inv WMaxLimPct RvrtTms	value	analog	normal	180
Inv WMaxLimPct Ena	value	analog	normal	1

Abbildung 72: Antwort Begrenzung der Einspeiseleitung bei Wirkleistungsvorgabe durch FWA

Der Testfall hat ergeben, dass die Begrenzung der Einspeiseleistung von dem PV-Controller richtig vorgenommen wird, um einer Abschaltung durch die $P_{AV,E}$ -Überwachung vorzugreifen. Bei einer Wirkleistungsvorgabe durch die FWA reagiert der PV-Controller korrekt und priorisiert die Vorgabe der FWA.

7.3.8 Auswertung Testfall 8: Reduzierung von Lastspitzen

Auf Grund des Problems mit der Datenabfrage der Stromzähler, wird in diesem Testfall ebenfalls der Netzbezug durch einen `Inject Node` auf dem PV-Controller vorgegeben. Der weitere Teil der Auswertung des Testfalls wird entsprechend des Testfalls 8 aus Abschnitt 7.2 durchgeführt. In der Konfiguration wird die Grenze zur Leistungsreduzierung

festgelegt. In diesem Fall wurde sie auf 1000 kW festgelegt. Bei einem Netzbezug oberhalb dieser Grenze, wird von dem PV-Controller die Variable `global.output.power_reduction` gesetzt. Unterhalb dieser Grenze wird diese wieder zurückgesetzt. Im Folgenden werden die Netzbezüge 1001 kW und 999 kW vorgegeben und es wird überprüft, ob die Variable `global.output.power_reduction` richtig gesetzt wird.

Tabelle 6: Ergebnis Testfall 9: Reduzierung der Lastspitzen

Netzbezug [kW]	Zustand <code>global.output.power_reduction</code>
1001	1
999	0

Die Ergebnisse der Überprüfung sind in der Tabelle 6 abgebildet. Dabei ist zu erkennen, dass eine Leistungsreduzierung richtig angezeigt wird.

7.3.9 Auswertung Testfall 9: Funktion der JEVIS-Datenbank

Entsprechend des Testfalls 9 aus Abschnitt 7.2 wird die Funktion der JEVIS-Datenbank überprüft. Auf Grund des Problems mit der Datenabfrage lassen sich nicht alle Register der angeschlossenen Geräte korrekt auslesen. Unabhängig davon werden in den korrekten Intervallen die abgefragten Daten von den `Intervall Nodes` an die `Logger Nodes` weitergereicht. Wenn ein Wert nicht richtig gelesen werden kann, wird dem `Logger Node` ein `undefined` übergeben. In der Abbildung 73 ist dies an dem `Logger Node` des Datenpunkt `Inv WattsHours` aus dem `Inverter 1 Flow` zu erkennen.

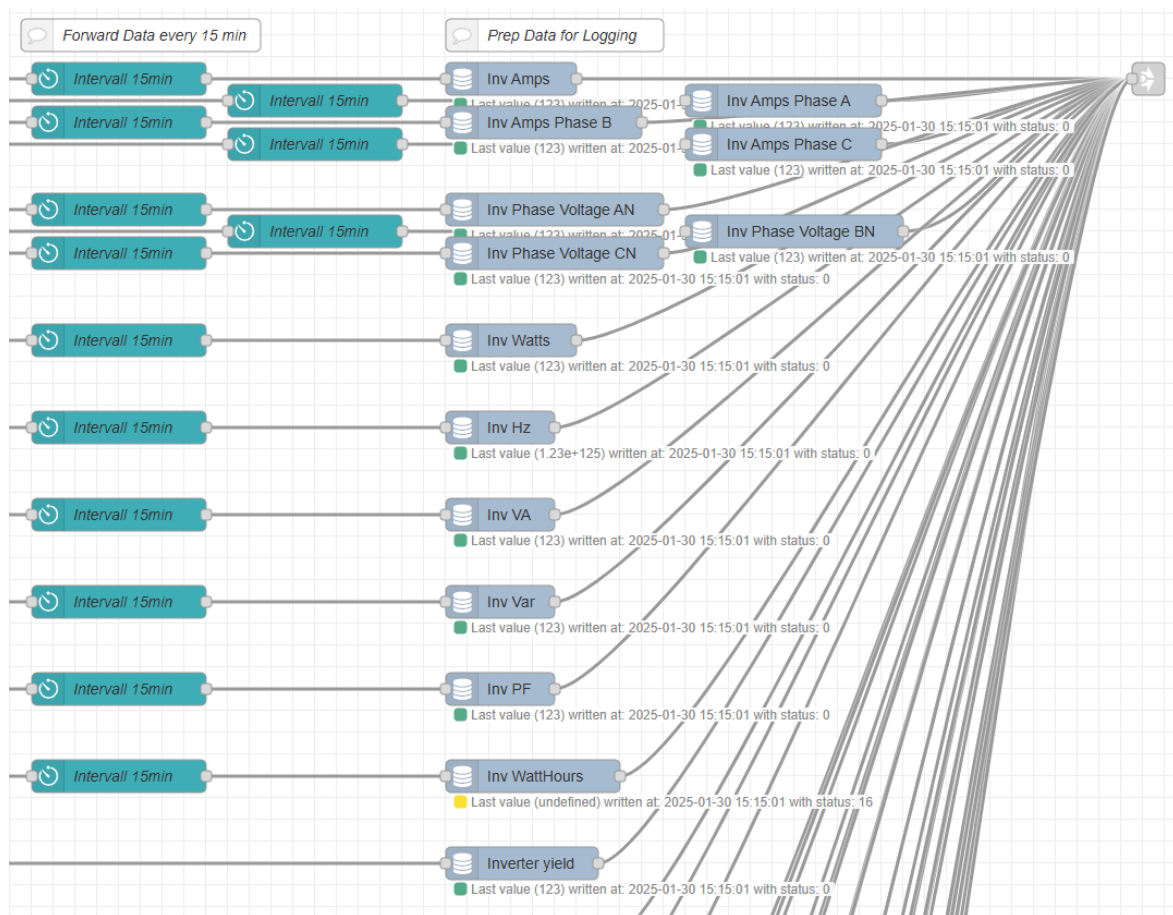


Abbildung 73: Intervall und Logger Nodes des Inverter 1 Flows

In den Logger Nodes wird die richtige Ringspeicherdauer festgelegt. Diese stimmt mit der in der Konfiguration übergebenen Dauer überein. Alle abgefragten Register werden in die Datenbank im Logging Flow, siehe Abbildung 30, übertragen – auch in dem Fall, dass ein Datenpunkt nicht korrekt eingelesen wurde.

7.4 Ergebniszusammenfassung

Die Erprobung hat gezeigt, dass für das korrekte Lesen und Schreiben der verbundenen Geräte ein Buffer Parser benötigt wird, um Werte, welche im 32-Bit-Datenformat angegeben werden, in der richtigen Form vorliegen zu haben. Dieser sollte im nächsten Entwicklungsschritt zur Marktreife implementiert werden. Des Weiteren wurde festgestellt, dass sowohl die Konfiguration als auch die Steuersignale der FWA, die Begrenzung von Lastspitzen, die Reduzierung der Einspeiseleistung und die Funktion der JEVIS-Datenbank ordnungsgemäß funktionieren.

Die aktuelle Erprobung ist eine reine Softwareerprobung zwischen zwei Controllern. Auf Grund der Wichtigkeit der korrekten Funktion des PV-Controllers, müssen mindestens zwei weitere Erprobungsphasen folgen. Dabei sollte eine Hardware in the Loop (HIL) Erprobung erfolgen. Bei der HIL-Erprobung wird der PV-Controller in einer Testumgebung mit realer Hardware erprobt, um sicherzustellen dass er korrekt arbeitet. Eine weitere Erprobung sollte anhand einer realen PV-Anlage, welche sich im Betrieb befindet, durchgeführt werden. Diese Anlage sollte jedoch auf Grund ihrer Größe keine FWA besitzen. Dadurch können FWA-Signale simuliert werden, ohne die Anforderungen an das Netzsicherheitsmanagement zu verletzen.

8 Fazit

Ziel dieser Arbeit war die Planung und Entwicklung eines plattformunabhängigen Controllers für die netzdienliche Steuerung und die Überwachung von industriellen Photovoltaikanlagen einschließlich der Konzeption und Realisierung eines standardisierten Tools für die projektspezifische Konfiguration.

Auf Basis der Anforderungsanalyse, welche anhand eines begleiteten Projekts durchgeführt wurde, wurden zahlreiche Anforderungen an den PV-Controller aus geltenden Normen und Gesetzen abgeleitet. Diese wurden bei der Konzeptionierung und der prototypischen Implementierung des PV-Controllers berücksichtigt. Für die Implementierung des PV-Controllers wurde ein Grundkonzept erarbeitet, welches zusätzlich die Anforderungen des betreuenden Unternehmens berücksichtigt. Die prototypische Umsetzung erfolgte auf einer als geeignet identifizierten Industrie-Hardware in Kombination mit der Open-Source-Programmierungsumgebung Node-RED. Dabei wurde die Node-RED Umgebung vor der Programmierung zur Konzeption der Programmstruktur unter Zuhilfenahme der zeichnerischen Funktionen genutzt. Es wurde stets auf die Plattformunabhängigkeit und die Erweiterbarkeit des PV-Controllers geachtet, so dass die Ergebnisse dieser Arbeit zukunftsicher, über viele Jahre, weiterverwendet werden können. Für die projektspezifische Konfiguration des PV-Controllers wurde ein standardisiertes Excel-Konfigurations-tool entwickelt. Die damit erstellte Konfigurationsdatei wird von dem PV-Controller bei Programmstart automatisch eingelesen, so dass der manuelle Konfigurationsaufwand auf ein Minimum reduziert wird. Lediglich ein kleiner Teil der Konfiguration muss händisch parametrisiert werden, wofür eine bebilderte Anleitung in dem Excel-Konfigurations-tool zur Verfügung steht.

Bei der Erprobung der Implementierung anhand der definierten Testfälle, wurde die wesentlichen Funktionen in der erstellten Testumgebung weitestgehend erfolgreich überprüft. Im Rahmen der Erprobung konnte festgestellt werden, dass Daten in 32-Bit-

Datenformaten nicht korrekt ausgelesen werden. Mehrere mögliche Lösungen für dieses Problem wurden benannt.

Eine besondere Herausforderung dieser Arbeit lag darin, dass der Umfang der Zielsetzung weitaus größer als erwartet war. Der hohe Aufwand resultierte vor allem daraus, dass sich neben der Konzeptionierung und der Entwicklung des PV-Controllers ein wesentlicher Mehraufwand aus der Entwicklung des Excel-Konfigurationstool sowie der Sichtung aller relevanten Normen und Gesetze zur Ableitung der regulatorischen Anforderungen ergab. Auch die Programmierung in Node-RED war umfangreicher als zunächst angenommen, da für das Energie-Monitoring eine Vielzahl weiterer Register von den Wechselrichtern und den Stromzählern abgefragt werden mussten und dabei stets auf die zukünftige Erweiterbarkeit geachtet wurde.

Für die Envidatec wurde die Grundlage für ein neues Produkt geschaffen. Nach weiteren Erprobungen kann der PV-Controller im Tagesgeschäft eingesetzt werden. Damit kann die bestehende Produktpalette zukünftig um einen plattformunabhängigen, preisattraktiven Controller erweitert werden. Der Open-Source-Gedanke von Node-RED passt zur Motivation, ein nachhaltiges und zukunftsfähiges Produkt zu entwickeln. Bei der Entwicklung konnten bereits von der Node-RED-Community erstellte Paletten verwendet werden, um die Programmierumgebung um weitere Funktionen zu erweitern. Bei der Weiterentwicklung des PV-Controllers können weitere von der Community erstellte Inhalte genutzt werden. Mit der Entwicklung eines zukunftsfähigen PV-Controllers wird zudem ein wichtiger Beitrag zur Energiewende und zur nachhaltigen Energieerzeugung geleistet.

9 Ausblick

Die Entwicklung des PV-Controllers ist noch nicht abgeschlossen, er wird in Zukunft weiterentwickelt und um neue Funktionen erweitert. So werden weitere Gerätetypen in das Excel-Konfigurationstool aufgenommen, so dass der PV-Controller diese ebenfalls

unterstützt. Die Anzahl der gleichzeitig angeschlossenen Geräte sollte angepasst werden, nachdem eine Marktanalyse durchgeführt wurde. Bevor der PV-Controller um neue Funktionen erweitert wird, wird die benötigte Node-RED Palette in das Programm hinzugefügt, damit alle Werte über Modbus korrekt übertragen werden. Im Anschluss folgen weitere Erprobungsphasen mit einem Testlauf an einem Live-System zur Sicherstellung der Funktion des PV-Controllers.

Die Grenze, bei der eine Reduzierung der Einspeiseleitung vorgenommen wird, sollte mit in das Konfigurationstool aufgenommen werden. Aktuell ist diese auf 20 kW unterhalb der maximalen Einspeiseleistung festgelegt. Gleichzeitig sollte überlegt werden, ob für die Regelung ein Hysteresekurve eingeführt wird, um ein mögliches Schwingen der Erzeugungsleistung zu verhindern.

Um eine vollständige Konfiguration über eine csv-Konfigurationsdatei vornehmen zu können, muss eine Möglichkeit geschaffen werden, die verwendeten Modbus Nodes über sogenannte Umgebungsvariablen zu konfigurieren. Dafür kann ein Änderungsvorschlag an der verwendeten Modbus Palette gemacht werden. Alternativ kann auch aktiv an der Entwicklung dieser Funktion mitgewirkt werden, damit diese schneller implementiert wird.

Die Signale zur Leistungsreduzierung und für das Not-Aus sollten beispielsweise über OPC UA anderen Controllern zugänglich gemacht werden. Alternativ könnten diese Signale auch direkt auf dem PV-Controller weiterverarbeitet werden. Das Not-Aus-Signal sollte jedoch unabhängig von der Bereitstellung für andere Controller auf dem PV-Controller umgesetzt werden. Dafür muss die Möglichkeit geschaffen werden, LS oder LaTS direkt mit dem PV-Controller anzusteuern.

Die Sicherheit des PV-Controllers sollte ebenfalls betrachtet werden. Die Anzahl an IP-Sockets, die sich mit dem PV-Controller verbinden können, sollte begrenzt werden, um den PV-Controller beispielsweise vor möglichen Distributed-Denial-of-Service-Angriffen oder unerwünschten Verbindungen zu schützen. Des Weiteren sollten stets sichere Passwörter verwendet werden, um einen unbefugten Zugriff zu verhindern.

Eine weitere denkbare Funktion wäre die Steuerung von Ladeinfrastruktur, um die Nutzung des selbstproduzierten Stroms zu erhöhen. Node-RED bietet zudem die Möglichkeit, das Open Charge Point Protocoll (OCPP) zu verwenden, um über eine standardisierte Schnittstelle mit der Ladeinfrastruktur zu kommunizieren. Dieses nutzt ebenfalls eine Ethernet-Verbindung. Bei Bedarf können auch weitere Schnittstellen implementiert werden. Dabei ist darauf zu achten, weiterhin plattformunabhängig zu bleiben.

Die Anforderungen an eine Zertifizierung zum Einsatz des PV-Controllers als eine Fernwirkunterstation der FWA könnten ermittelt werden, um den Controller direkt als Teil der FWA einzusetzen. Dadurch wird die benötigte Hardware zur Realisierung des Netzsicherheitsmanagement weiter reduziert.

Zur Dokumentation der Änderungen am Programm bei der Weiterentwicklung des PV-Controllers sollte ein Changelog angefangen werden, in dem alle Änderungen dokumentiert werden. Durch die zukünftigen Erweiterungen des PV-Controllers kann die Heap-Size von Node-RED zu klein werden und muss bei Bedarf angepasst werden, um eine performante Funktion des PV-Controllers sicherzustellen.

Es ist zu erwarten, dass der Ausbau von industriellen PV-Anlagen zunehmen wird und die Nachfrage nach Controllern für die Steuerung von PV-Anlagen und das Energie-Monitoring steigen wird. So könnten Kunden der Envidatec diese vermehrt nachfragen.

Durch zukünftige Erweiterungen und den Einsatz des PV-Controllers bei Kunden ist zu erwarten, dass durch den PV-Controller CO₂ eingespart wird und ein wichtiger Beitrag zur Energiewende geleistet wird.

Literaturverzeichnis

- [1] BMZ, „Folgen des Klimawandels - Bundesministerium für Wirtschaftliche Zusammenarbeit und Entwicklung,“ 25.09.2023. [Online]. Available: <https://www.bmz.de/de/themen/klimawandel-und-entwicklung/folgen-des-klimawandels-124774>. [Zugriff am 26.11.2024].
- [2] ABB, „ABB - Energiemonitoring Transparenz schaffen für Energie-, Betriebs- und Wartungskosten – jederzeit und überall,“ ABB, [Online]. Available: <https://new.abb.com/low-voltage/de-ch/loesungen/energiemonitoring>. [Zugriff am 01.02.2025].
- [3] T. Frank, Interviewee, *Geschäftsführer Envidatec GmbH, Gespräch zu den Design-Vorgaben des Unternehmens*. [Interview]. 30.11.2024.
- [4] Envidatec, „Envidatec - Energieberatung im Dienste der Nachhaltigkeit,“ 2024. [Online]. Available: <https://envidatec.com/>. [Zugriff am 26.11.2024].
- [5] Envidatec, „JEVis,“ 2024. [Online]. Available: <https://jevis.de/>. [Zugriff am 26.11.2024].
- [6] Sun and Charge, „Sun and Charge - Solarstrom der Bewegt,“ 2024. [Online]. Available: <https://sunandcharge.com/>. [Zugriff am 26.11.2024].
- [7] N. Heinrich, Interviewee, *Geschäftsführer Sun and Charge Germany GmbH / Leiter Engineering & Automation Envidatec GmbH, Folgegespräch zu den Design-Vorgaben des Unternehmens*. [Interview]. 03.12.2024.

- [8] Waveshare, „Waveshare - RS485 TO ETH (B) User Manual,“ [Online]. Available: <https://files.waveshare.com/upload/4/4d/RS485-to-eth-b-user-manual-EN-v1.33.pdf>. [Zugriff am 28 11 2024].
- [9] VDE, „VDE FNN Vereinfachter Anschluss und Nachweis von Erzeugungsanlagen und Speichern mit Netzanschluss in der Mittel- und Hochspannung,“ 3 2024. [Online]. Available: <https://www.vde.com/resource/blob/2306364/ae50ed2dfe20e6584fad16e2525565e2/nelev-hinweis-download-data.pdf>. [Zugriff am 28 11 2024].
- [10] „Verordnung über technische Anforderungen an Energieanlagen (Energieanlagen-Anforderungen-Verordnung - EAAV),“ 16 05 2024. [Online]. Available: https://www.gesetze-im-internet.de/eaav/_3.html. [Zugriff am 03 12 2024].
- [11] VDE, VDE-AR-N 4105:2018-11, VDE Verlag, 2018-11.
- [12] Ziehl, „Ziehl - Betriebsanleitung EFR4001IP,“ 08 05 2024. [Online]. Available: https://www.ziehl.com/media/f9/46/9a/1715674072/EFR4001IP_BA_ZIEHL_2024-05-08_DE.pdf. [Zugriff am 22 01 2025].
- [13] SMA, „Anleitung zum FNN-Hinweis „Vereinfachter Anschluss und Nachweis von Erzeugungsanlagen und Speichern mit Netzanschluss in der Mittel- und Hochspannung“,“ 16 12 2024. [Online]. Available: <https://files.sma.de/downloads/NELEV-GG10-CORE2-EDMM-20-TI-de-11.pdf>. [Zugriff am 6 1 2025].
- [14] SMA, „Technische Information - SUNNY TRIPOWER CORE2 - Parametereinstellungen zur Einhaltung der Netzanschlussrichtlinien,“ [Online]. Available: https://files.sma.de/downloads/Paraeinst_Netzanschluss_CORE2-TI-de-13.pdf. [Zugriff am 22 01 2025].

- [15] BDEW Bundesverband der Energie- und Wasserwirtschaft e.V. Landesgruppe Norddeutschland, „TAB NS Nord 2023 v2.0,“ 15 5 2024. [Online]. Available: <https://www.wemag-netz.de/media/40/download>. [Zugriff am 2 12 2024].
- [16] WEMAG Netz GmbH, „TAB Mittelspannung - Technische Anschlussbedingungen für den Anschluss an das Mittelspannungsnetz der WEMAG NETZ GmbH,“ 1 1 2024. [Online]. Available: <https://www.wemag-netz.de/media/375/download?inline>. [Zugriff am 16 12 2024].
- [17] VDE, VDE-AR-N 4110:2023-09, VDE Verlag, 2023-09.
- [18] VDE, „VDE | FNN TAR Mittelspannung - Formulare Anhang E der VDE-AR-N 4110:2018-11,“ 11 2018. [Online]. Available: <https://www.vde.com/resource/blob/1798770/5630224a6a17c4a2ce11867ed66fd77b/vde-ar-n-4110-formulare-anhang-e-data.pdf>. [Zugriff am 01 02 2025].
- [19] „Elektrotechnische-Eigenschaften-Nachweis-Verordnung - NELEV § 2 Nachweis der Einhaltung der allgemeinen technischen Mindestanforderungen von Erzeugungsanlagen,“ 16 5 2024. [Online]. Available: https://www.gesetze-im-internet.de/nelev/_2.html. [Zugriff am 29 11 2024].
- [20] Bundesministerium für Wirtschaft und Klimaschutz (BMWK), „BMWK - Solarpaket 1,“ 2024. [Online]. Available: <https://www.bmwk.de/Redaktion/DE/Artikel/Energie/solarpaket-1.html>. [Zugriff am 3 12 2024].
- [21] WEMAG Netz GmbH, „Richtlinie RL N 04 „Technische Umsetzung des Einspeisemanagements bei der WEMAG Netz GmbH“,“ 09 03 2022. [Online]. Available: <https://www.wemag-netz.de/media/59/download?inline>. [Zugriff am 2 12 2024].

- [22] Lambrecht meteo, „sun_e_Betriebsanleitung,” [Online]. Available: https://www.lambrecht.net/upload/manuals/sun_e_Betriebsanleitung.pdf. [Zugriff am 18 12 2024].
- [23] WEMAG Netz GmbH, „WEMAG Netz GmbH - Netznutzung Strom,” [Online]. Available: <https://www.wemag-netz.de/netznutzung-strom#gesch%C3%A4ftskunden>. [Zugriff am 22 01 2025].
- [24] OpenJS Foundation & Contributors, „Node-RED - About,” [Online]. Available: <https://nodered.org/about/>. [Zugriff am 10 12 2024].
- [25] OpenJS Foundation & Contributors, „Node-RED - @envidatec/interval,” [Online]. Available: <https://flows.nodered.org/node/@envidatec/interval>. [Zugriff am 17 01 2025].
- [26] OpenJS Foundation & Contributors, „Node-RED - @envidatec/node-red-contrib-logger,” [Online]. Available: <https://flows.nodered.org/node/@envidatec/node-red-contrib-logger>. [Zugriff am 17 01 2025].
- [27] OpenJS Foundation & Contributors, „Node-RED,” [Online]. Available: <https://nodered.org/>. [Zugriff am 22 01 2025].
- [28] OpenJS Foundation & Contributors, „Node-RED - Running Node-RED locally,” [Online]. Available: <https://nodered.org/docs/getting-started/local>. [Zugriff am 4 12 2024].
- [29] Wago, „WAGO Edge Devices IT and OT in One Device,” 23 11 2023. [Online]. Available: https://www.wago.com/us/d/Info_60465863. [Zugriff am 14 01 2025].
- [30] Siemens, „Lern-/Lehrunterlagen - Siemens Automation Cooperates with Education - TIA Portal Modul 094-100 Node-RED mit SIMATIC IOT2000,” 6 2019. [Online]. Available: <https://www.automation.siemens.com/sce-static/learning-training->

documents/tia-portal/advanced-communication/sce-094-100-node-red-iot2000-de.pdf?ref=learn.umh.app. [Zugriff am 14 01 2025].

- [31] OpenJS Foundation & Contributors, „Node-RED - Working with messages,“ [Online]. Available: <https://nodered.org/docs/user-guide/messages>. [Zugriff am 08 01 2025].
- [32] SMA, „SMA-SUNNY TRIPOWER CORE2 STP 110-60 Datenblatt,“ 11 2023. [Online]. Available: <https://files.sma.de/downloads/STP110-60-AFCI-DS-de-21.pdf>. [Zugriff am 11 10 2024].
- [33] WAGO, „Controller PFC200 (750-8212/025-002) | WAGO,“ [Online]. Available: https://www.wago.com/de/sps/controller-pfc200/p/750-8212_025-002. [Zugriff am 17 12 2024].
- [34] Janitza, „Janitza - MID-Energiezähler Produkthandbuch,“ 04 2018. [Online]. Available: <https://www.janitza.de/files/download/manuals/current/B-Series/janitza-bhb-b2x-de.pdf>. [Zugriff am 14 01 2025].
- [35] KEBA, „KeContact Wallboxen | Übersicht Ladestationen | KEBA,“ [Online]. Available: <https://www.keba.com/de/emobility/products/product-overview/wallboxen>. [Zugriff am 20 01 2025].
- [36] Compleo, „Compleo Quickguide eBOX Modbusenergiemanagement,“ 14 08 2023. [Online]. Available: https://www.compleo-charging.com/fileadmin/Documentcenter/Modbus_eBOX/DE_Quickguide_eBOX_-_Modbus_Energiemanagement_20230814.pdf. [Zugriff am 17 10 2024].
- [37] KUNBUS, „Revolution Pi - The Industrial Pi,“ [Online]. Available: https://revolutionpi.com/fileadmin/downloads/flyer/revolution-pi-flyer_de.pdf. [Zugriff am 12 10 2024].

- [38] SFERALABS, „Strato Pi - The Industrial Raspberry Pi -Sferalabs,“ unbekannt. [Online]. Available: <https://sferalabs.cc/strato-pi/>. [Zugriff am 7 10 2024].
- [39] KUNBUS, „Industrie PC auf Raspberry Pi Basis Überblick | Industrial Raspberry Pi - Revolution Pi,“ [Online]. Available: <https://revolutionpi.com/de/revolution-pi-serie>. [Zugriff am 7 12 2024].
- [40] KUNBUS, „Revolution Pi Online Shop,“ [Online]. Available: <https://revolutionpi.com/shop/de>. [Zugriff am 17 01 2025].
- [41] KUNBUS, „Steuerung Kompakt REVPI CORE Schlank. Modular. Kosteneffizient,“ [Online]. Available: <https://revolutionpi.com/de/revpi-core>. [Zugriff am 7 12 2024].
- [42] Modbus Organization, Inc., „MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3,“ 26 04 2012. [Online]. Available: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. [Zugriff am 11 12 2024].
- [43] Modbus Organization, Inc., „MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b Kapitel 4.4.1.2 Seite 23,“ 24 10 2006. [Online]. Available: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf. [Zugriff am 11 12 2024].
- [44] OpenJS Foundation & Contributors, „Node-RED - node-red-contrib-modbus,“ [Online]. Available: <https://flows.nodered.org/node/node-red-contrib-modbus>. [Zugriff am 17 01 2025].
- [45] SMA, „Betriebsanleitung - SUNNY TRIPOWER CORE2 - STP110-60-BE-de-16.pdf,“ 1 3 2024. [Online]. Available: <https://files.sma.de/downloads/STP110-60-BE-de-16.pdf>. [Zugriff am 18 12 2024].

- [46] SMA, „SUNNY TRIPOWER CORE2 (STP 110-60) - SunSpec Modbus Overview,“ 28 05 2024. [Online]. Available: https://files.sma.de/downloads/CORE2_STP_110-60_Sunspec_Modbus_Model_PICS_Overview_V12.zip. [Zugriff am 16 12 2024].
- [47] SMA, „Technische Information SMA und SunSpec Modbus-Schnittstelle Datenblatt,“ 17 5 2018. [Online]. Available: <https://files.sma.de/downloads/SMA-Modbus-general-TI-de-10.pdf>. [Zugriff am 11 12 2024].
- [48] SunSpec Alliance, „SUNSPEC SPECIFICATIONS AND INFORMATION MODELS,“ [Online]. Available: <https://sunspec.org/specifications/#tabs|1>. [Zugriff am 07 01 2024].
- [49] SunSpec Alliance, „Product Certification Registry,“ 2025. [Online]. Available: <https://sunspec.org/certified-registry/>. [Zugriff am 08 01 2025].
- [50] Github, „Github: MU-Envidatec/Bachelorarbeit_MU,“ [Online]. Available: https://github.com/MU-Envidatec/Bachelorarbeit_MU. [Zugriff am 2 12 2024].
- [51] OpenJS Foundation & Contributors, „Node-RED - Node credentials,“ [Online]. Available: <https://nodered.org/docs/creating-nodes/credentials>. [Zugriff am 18 12 2024].
- [52] Loytec, „Details für L-INX Configurator 8.2.8 (64bit),“ 30 08 2024. [Online]. Available: [https://www.loytec.com/de/support/download/doc_details/2324-L-INX-Configurator-7.6.0-\(64-bit\)](https://www.loytec.com/de/support/download/doc_details/2324-L-INX-Configurator-7.6.0-(64-bit)). [Zugriff am 20 12 2024].
- [53] elbar, „SOURCEFORGE / QModMaster Modbus TCP,RTU Master Software,“ 15 08 2022. [Online]. Available: <https://sourceforge.net/projects/qmodmaster/>. [Zugriff am 13 1 2025].

[54] OpenJS Foundation & Contributors , „Node-RED - node-red-contrib-buffer-parser,“
[Online]. Available: [https://flows.nodered.org/node/node-red-contrib-buffer-
parser](https://flows.nodered.org/node/node-red-contrib-buffer-parser). [Zugriff am 28 01 2025].

Anhang

Anhang A: Blatt E.8

Dieses Dokument stammt aus der Quelle [18] und ist als Anhang A auf der beiliegenden DVD zu finden.

Anhang B: Blatt E.9

Dieses Dokument stammt aus der Quelle [18] und ist als Anhang B auf der beiliegenden DVD zu finden.

Anhang C: Single-Line-Diagram

Dieses Dokument ist auf der beiliegenden DVD als Anhang C zu finden.

Anhang D: Quellcode

Der Quellcode der Node-RED Programmierung ist auf der beiliegenden DVD als Anhang D zu finden.

Im Anhang D ist zudem eine kleine Anleitung beigefügt, wie sich der Quellcode in Node-RED importieren lässt.

Anhang E: Konfigurationsdatei

Eine Beispiel Konfigurationsdatei ist auf der DVD im Anhang F zu finden.

Anhang F: Excel-Konfigurationstool

Das Excel-Konfigurationstool ist auf der beigefügten DVD als Anhang E zu finden.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____