

Bachelorarbeit 2015

Evaluation of the photovoltaic power plant of the GridLab



Student : Kevin Wyer
Dozent : Dominique Genoud
Abgabedatum : 27. Juli 2015
Webseite : www.hevs.ch

Management Summary

In der vorliegenden Bachelorarbeit wird eine Software beziehungsweise ein Überwachungssystem für die Fachhochschule HES-SO in Sion implementiert. Mithilfe dieses Überwachungssystems kann die produzierte Energie der Photovoltaikanlage am Gebäude der HES-SO Sion grafisch angezeigt werden.

Die Arbeit umfasst folgende vier Phasen:

- Recherchephase
- Analysephase
- Implementierungsphase
- Testphase

Diese Phasen widerspiegeln auch das Vorgehen sowie die Ziele der Arbeit. Sie zeigen den Ablauf für die Entwicklung der Software. In der ersten Phase wurden Recherchen durchgeführt, um die Software kennenzulernen. Anschliessend wurde die bereits bestehende Software analysiert. Der Hauptteil umfasste dann die Implementierung der Software wie auch die Zusammenführung mit einer zweiten Anlage, welche schrittweise erklärt wurden. Die Testphase verlief parallel zur Implementierungsphase, da die neuen Funktionen jeweils direkt auf ihre Funktionsfähigkeit getestet wurden.

Die Zusammenführung beider Anlagen war erfolgreich und die Software kann nun angewendet werden. Es ist möglich, die Energiegenerierung in einer Tages-, Monats- und Jahreskurve anzuzeigen. Bei jeder Kurve ist jeweils auch ein Vergleich mit zwei frei wählbaren Zeitintervallen möglich. So können beispielsweise bei der Tageskurve zwei weitere frei wählbare Tage mit dem aktuellen Tag verglichen werden. Durch die Verknüpfung der Software mit der Meteostation kann zudem die theoretisch mögliche Energieproduktion angezeigt werden. Überdies kann durch den geografischen Standort der Photovoltaikanlage berechnet werden, wie gross die generierte Energie im Jahr sein sollte. Bei allfälligen Problemen an der Anlage gibt es ausserdem ein Register, in welchem der Fachmann detailliertere Informationen über Strom und Spannung der Anlage erhält, um das Problem ausfindig zu machen.

Aufgewendete Stunden für die Bachelorarbeit: 401 Stunden (ohne Vorbereitung für die Verfechtung).

Schlüsselbegriffe: GridLab, LabView, Photovoltaik, grafische Programmierung

Vorwort und Dank

Die Bachelorarbeit wurde im Rahmen des Moduls Bachelorarbeit in Zusammenarbeit mit der Fachhochschule HES-SO in Sion geschrieben. Das Thema wurde von Christoph Ellert vorgegeben. Herr Ellert ist darum der Auftraggeber, welcher den Auftrag erteilte, eine Software zur Messung der Energiegenerierung einer Photovoltaikanlage zu entwickeln.

Ich habe das vorgeschlagene Thema gewählt, da es mich unmittelbar interessierte. Ausschlaggebend war vor allem, da es sich um etwas total Neues für mich handelte und ich diese Herausforderung mag und gerne Neues erlerne. Vor dieser Arbeit hatte ich noch nie grafisch programmiert. Zudem gewinnt die Energiegewinnung über Solarenergie immer mehr an Bedeutung. Photovoltaikanlagen stellen eine nachhaltige Technologie dar, welche für die Zukunft sehr wichtig ist.

Das Ziel der Software ist es, die gesamterworbene Energie der Photovoltaikanlage der HES-SO in Sion mittels Diagrammen grafisch anzuzeigen. Dementsprechend geht es darum, diese Software mithilfe eines vorgegeben Programmes zu implementieren.

Die Schwierigkeit meiner Arbeit lag in der Implementierung der Datenbankverbindung, da diese viel mehr Zeit in Anspruch nahm als geplant. Grund dafür war, dass im Gegensatz zum alten Programm eine andere Datenbank benutzt werden musste, wodurch viele zeitintensive Anpassungen vorgenommen werden mussten. Durch diese Zeiteinbusse war zum Schluss nicht mehr möglich, das Management der Fehlermeldungen zu implementieren.

Als erstes bedanke ich mich herzlich bei Ivan Bittel, der sich die Zeit genommen hat, mir bei allfälligen Problemen mit dem Programm LabView weiterzuhelfen. Ein weiterer Dank geht an meinen Betreuer, Herrn Dominique Genoud, welcher mich während der Arbeit stets unterstützt hat und bei Fragen weiterhalf. Des Weiteren bedanke ich mich bei Herrn Michael Clausen, welcher mir bei den Schwierigkeiten mit der Datenbankverbindung weiterhelfen konnte. Ein letzter Dank geht an Herrn Christoph Ellert, der mich stets unterstützt hat und bei Fragen zur Verfügung stand.

Ich weise darauf hin, dass in der vorliegenden Arbeit auf die Formulierung der weiblichen Form verzichtet wird, um die Lesbarkeit zu erhöhen. Die Frauen sind in der männlichen Form miteinbezogen.

Inhaltsverzeichnis

Management Summary.....	ii
Vorwort und Dank	iii
Inhaltsverzeichnis	iv
Tabellenverzeichnis	vi
Abbildungsverzeichnis	vii
Abkürzungsverzeichnis	ix
1. Einleitung	1
1.1 Ausgangslage.....	1
1.2 Problemstellung.....	1
1.3 Zielsetzung	1
1.4 Methodik	2
1.5 i3 JSON Toolkit.....	3
2. Theoretischer Hintergrund zu Photovoltaik.....	3
2.1 Die Solaranlage	4
2.1.1 Funktionsweise der Solarzelle	4
2.1.2 Funktionsweise der Solaranlage.....	5
2.2 Schweizer Anteil an Solarenergie	6
3. Porträt der PV-Anlage in Sion	7
4. Recherchephase	8
4.1 LabView 2014.....	8
4.2 Photovoltaik	9
5. Analysephase.....	9
6. Implementierungsphase	10
6.1 Register Solarmax	11
6.1.1 Verbindungsaufbau zur Datenbank influxDB.....	12
6.1.2 Daten von beliebigem Datum laden	15
6.1.3 Verarbeitung vom JSON String und speichern in .csv-Datei	16

6.1.4	Daten aus .csv-Datei lesen und in einem Graphen anzeigen.....	20
6.2	Register Summierungsmodul.....	32
6.2.1	Summierungsmodul Tageskurve	33
6.2.2	Summierungsmodul Monatskurve.....	36
6.2.3	Summierungsmodul Jahreskurve.....	37
6.3	Register Fachmann	37
7.	Testphase	39
8.	Schlussfolgerung.....	40
	Literaturverzeichnis.....	41
	Anhang I: Daten der Bachelorarbeit	42
	Anhang II: Daily report	44
	Anhang III: product backlog	48
	Selbständigkeitserklärung.....	58

Tabellenverzeichnis

Tabelle 6.1: Beschreibung der Elemente für die Datenbankverbindung	14
--	----

Abbildungsverzeichnis

Abbildung 1.1: i3 JSON Toolkit	3
Abbildung 2.1: Funktionsweise der Solarzelle	4
Abbildung 2.2: Funktionsweise der Solaranlage.....	5
Abbildung 2.3: In der Schweiz produzierte Energie	6
Abbildung 3.1: Übersicht der PV-Anlagen der HES-SO in Sion.....	7
Abbildung 4.1: Beispielprogramm LabView Rechner	8
Abbildung 6.1: Register der Software.....	10
Abbildung 6.2: Flussdiagramm Programmablauf.....	10
Abbildung 6.3: Flussdiagramm Solarmax.....	11
Abbildung 6.4: vom Solarpanel zur .csv-Datei.....	12
Abbildung 6.5: Darstellung JSON Objekt	13
Abbildung 6.6: Aufbau und Test der Datenbankverbindung	13
Abbildung 6.7: Ergänzte Datenbankabfrage mit gewähltem Datum.....	15
Abbildung 6.8: Konvertierung von Datum 1.9.2014 nach 01.09.2014.....	16
Abbildung 6.9: Resultat Konvertierung von Datum 1.9.2014 nach 01.09.2014.....	16
Abbildung 6.10: JSON String aus Datenbank	17
Abbildung 6.11: Verarbeitung JSON String	18
Abbildung 6.12: Eintrag mit UNIX Zeitstempel	18
Abbildung 6.13: Eintrag mit LabView Zeitstempel	18
Abbildung 6.14: Eintrag mit Zeitstempel Sekunden seit Mitternacht.....	19
Abbildung 6.15: Daten in .csv-Datei speichern.....	19
Abbildung 6.16: Verarbeiteter JSON String in Solarmax.csv mit Editor	19
Abbildung 6.17: Verarbeiteter JSON String in Solarmax.csv mit Microsoft Excel	19
Abbildung 6.18: Überblick des gesamten Programmes.....	20
Abbildung 6.19: Schema Solarmax.....	22
Abbildung 6.20: Die Tageskurve	23
Abbildung 6.21: Die Monatskurve	24
Abbildung 6.22: Jahresdaten 2015_00_00_Solarmax.csv.....	25
Abbildung 6.23: Soll-Werte Juni, Mai & April.....	25
Abbildung 6.24: Jahresdaten bis Mai	26
Abbildung 6.25: Jahreskurve bis Mai	26
Abbildung 6.26: Jahreskurve am 01. Juli.....	27
Abbildung 6.27: Jahresdaten am 01. Juli	27
Abbildung 6.28: Daten vom aktuellen Tag laufend aktualisieren	28

Abbildung 6.29: Simulationsdaten anzeigen	30
Abbildung 6.30: Meteokurve im Tagesgraph	31
Abbildung 6.31: Diagramm Summierungsmodul	32
Abbildung 6.32: Summierungsmodul Übersicht.....	33
Abbildung 6.33: Summierungsmodul Logik aktueller Tag.....	34
Abbildung 6.34: Summierungsmodul Tageskurve aktueller Tag.....	35
Abbildung 6.35: Summierungsmodul Tageskurve beliebiges Datum.....	36
Abbildung 6.36: Summierungsmodul Monatskurve	36
Abbildung 6.37: Summierungsmodul Jahreskurve	37
Abbildung 6.38: Schema Solarmax	38
Abbildung 6.39: Übersicht Register Fachmann	38

Abkürzungsverzeichnis

AC	Alternating Current (Wechselstrom)
AKW	Atomkraftwerk
bzw.	beziehungsweise
DB	Datenbank
DC	Direct Current (Gleichstrom)
d.h.	das heisst
JSON	JavaScript Object Notation
kWh	Kilowattstunde
PV	Photovoltaik
UI	User Interface
usw.	und so weiter
z.B.	zum Beispiel

1. Einleitung

1.1 Ausgangslage

Die vorliegende Bachelorarbeit wird in Zusammenarbeit mit der Fachhochschule HES-SO in Sion geschrieben. Es wird auf die Thematik „Evaluation of photovoltaic power plant“ eingegangen, wobei die produzierte Energie der Photovoltaikanlage (im Weiteren PV-Anlage) in Sion grafisch erfasst werden soll. Die Erfassung der Daten stammen von zwei verschiedenen Anlagen, dem DC-Bus und dem Solarmax, wobei erstere von Ivan Bittel, Student Systemtechnik in Sion, implementiert wird.

1.2 Problemstellung

Gemäss Christoph Ellert wurde in der Vergangenheit bereits ein Programm implementiert, welches jedoch nicht den Anforderungen entsprach und nur die Anlage DC-Bus abdeckte. Das User Interface (Benutzeroberfläche) war unübersichtlich gestaltet, was die Benutzerfreundlichkeit stark beeinträchtigte.

1.3 Zielsetzung

Das Ziel dieser Arbeit ist, mithilfe des Programmes LabView ein Überwachungssystem zu entwickeln, um die generierte Energie der PV-Anlage grafisch darzustellen. Da dieses System von zwei Studenten implementiert wird, ist es wichtig, dass die beiden erarbeiteten Teile schliesslich mittels eines sogenannten Summierungsmoduls zusammengeführt werden.

Das Überwachungssystem ist während der Entwicklung und Implementierung durchgehend in Betrieb, damit fortlaufend Fehler an der Anlage oder in der Programmierung erkannt werden können und entsprechend reagiert werden kann.

1.4 Methodik

Dieser Abschnitt befasst sich mit dem methodischen Vorgehen der Bachelorarbeit. Das Vorgehen lässt folgende Gliederung zu:

1. Recherchephase
2. Analysephase
3. Implementierungsphase
4. Testphase

In der ersten Phase ging es vor allem darum, mittels Online Tutorials das Programm LabView kennenzulernen. In der Analysephase wurde das überarbeitete Überwachungssystem von Ivan Bittel untersucht, um die komplexeren Funktionen besser zu verstehen, da in den Tutorials vor allem Grundfunktionen erklärt wurden. In der Implementierungsphase wurde dann, basierend auf dem bestehenden System, das Überwachungssystem für die Anlage Solarmax implementiert. Die Testphase wird hier als vierte Phase aufgeführt, welche jedoch hauptsächlich parallel zur Implementierungsphase verlief, da die implementierten Funktionen fortlaufend getestet wurden.

Die Planung der vorliegenden Arbeit erfolgte mit Hilfe der agilen Methodik SCRUM. Dazu wurde in Zusammenarbeit mit Herrn Christoph Ellert der Anforderungskatalog, auch genannt Product Backlog, verfasst, damit eine Abgrenzung für den Arbeitsumfang vorhanden ist(siehe Anhang III). Daraufhin wurde die Planung erstellt, welche im Anhang II zu finden ist.

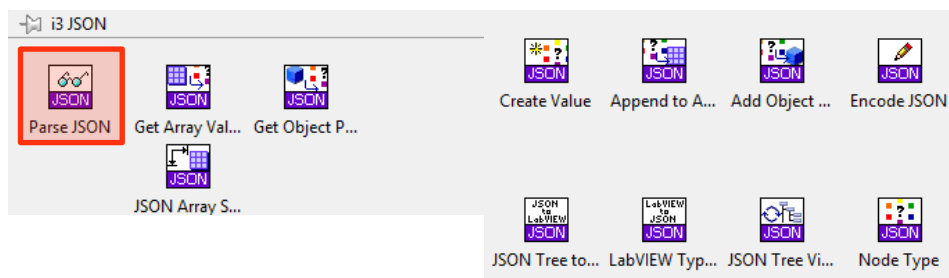
Die Bachelorarbeit wurde dann in fünf Iterationen (sprints) durchgeführt, welche jeweils zwei Wochen dauerten. Ende eines jeden Sprints war wichtig, mit dem product owner (Christoph Ellert) die Funktionsfähigkeit der Funktionen zu testen.

Alle Abbildungen dieser Arbeit wurden vom Programm LabView entnommen, wodurch die Quelle im Weiteren nicht dauernd wiederholt wird. Stammt eine Abbildung nicht aus LabView, wird die Quelle explizit angegeben.

1.5 i3 JSON Toolkit

Erste Probleme traten bereits bei der Datenbankverbindung auf. Das bestehende Programm deckte nur den DC-Bus ab, wodurch die Implementierung nur an die MySQL Datenbank angepasst war. Da die Daten, die aus dem Solarmax ausgelesen werden, in eine influxDB gespeichert werden, musste der Verbindungsaufbau zur Datenbank ganz anders erstellt werden. Dafür wurde ein Meeting mit Herrn Michael Clausen arrangiert, um die Vorgehensweise zu besprechen. Es resultierte, dass im LabVIEW die Datenbankverbindung mithilfe eines Webservices hergestellt werden muss, der die Daten mit einem http GET als JSON in einer .txt-Datei abspeichert. Um anschliessend mit diesem Format weiterzuarbeiten, musste das i3 JSON v1.0.0.10 Toolkit im JKI VI Package Manager heruntergeladen werden, um zusätzlich folgende Indikatoren zu erhalten:

Abbildung 1.1: i3 JSON Toolkit



Nachdem das Toolkit heruntergeladen und die Funktionsweise getestet wurden, fiel auf, dass das Element Parse JSON (Abbildung 1.1 rot markiert) nicht so funktioniert, wie es sollte. Nach Absprache mit Herrn Aldo Vaccari stellte sich heraus, dass die Konvertierung auf einem anderen Weg durchgeführt werden muss, welche in Kapitel 6.1.3 erläutert wird.

2. Theoretischer Hintergrund zu Photovoltaik

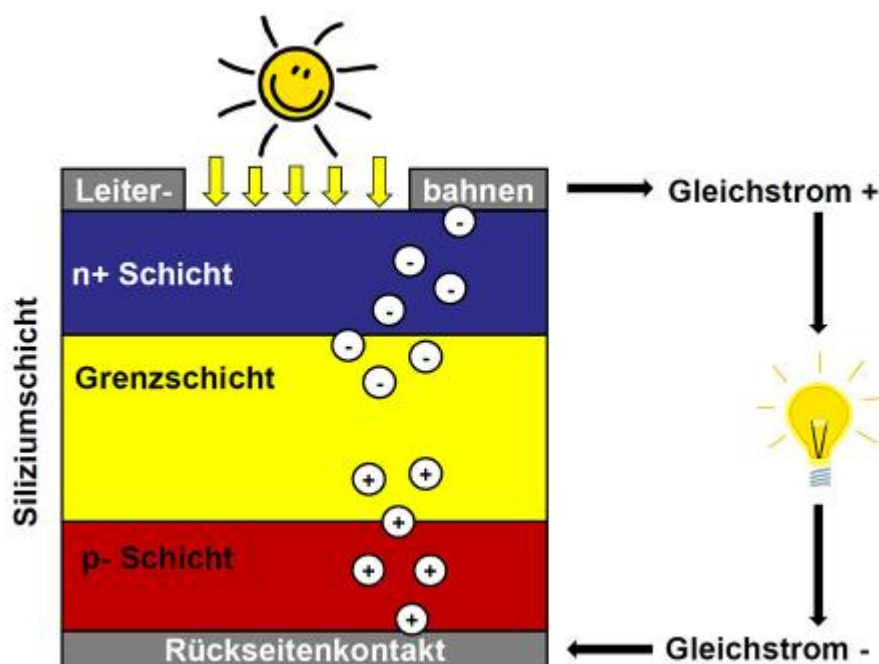
In diesem Kapitel wird die Photovoltaik erklärt, damit die nötigen Grundkenntnisse zur Verständlichkeit dieser Arbeit vorhanden sind.

2.1 Die Solaranlage

„Eine Solaranlage besteht aus vielen einzelnen Solarzellen, die etwa so gross sind wie eine Handfläche. Eine einzelne Solarzelle produziert nur wenig elektrischen Strom. Um einen MP3-Player zu betreiben, bräuchtest du 7 solcher Solarzellen. Meistens sind 60 Solarzellen in einer Platte zusammengefasst. Eine solche Platte heisst Solarmodul. Eine Photovoltaikanlage auf einem Hausdach besteht aus 10 bis 100 solcher Solarmodule“ (Centralschweizerische Kraftwerke AG, 2014).

2.1.1 Funktionsweise der Solarzelle

Abbildung 2.1: Funktionsweise der Solarzelle



Quelle: (Centralschweizerische Kraftwerke AG, 2014)

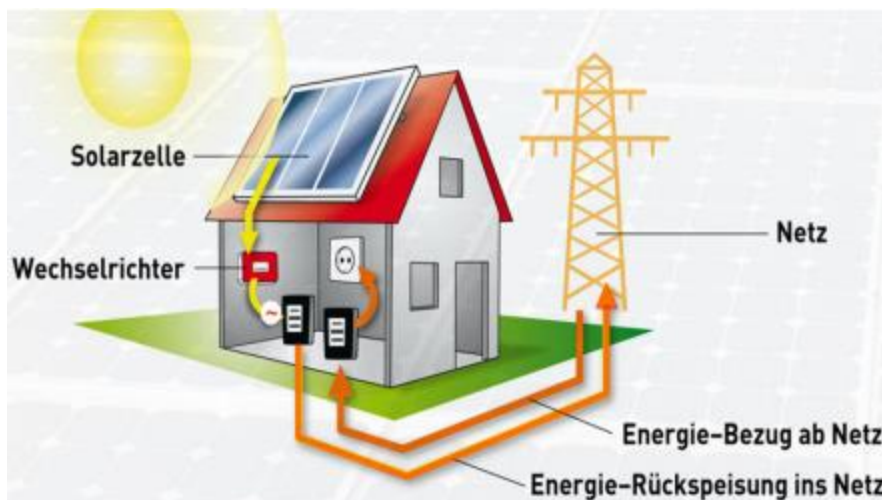
Um die Sonnenstrahlen in Strom umzuwandeln, verwenden wir eine spezielle Eigenschaft der Sonne. Im Sonnenlicht sind unzählige winzige Energieträger vorhanden, sogenannte Photonen. Wenn diese Photonen auf bestimmte Stoffe wie z.B. Silizium treffen, werden von ihnen negativ geladene Elektronen freigesetzt. Die negativen Elektronen wandern anschliessend nach oben zu den Leiterbahnen. Gleichzeitig wandern die „Löcher“, die positiv geladen sind, zum Rückseitenkontakt. (Centralschweizerische Kraftwerke AG, 2014)

Positive und negative Ladungen ziehen sich an. Doch da die direkte Verbindung durch die Grenzschicht verhindert wird, nehmen die Elektronen einen „Umweg“ über die angeschlossene Leitung, wobei sie z.B. eine Lampe zum Leuchten bringen. (Centralschweizerische Kraftwerke AG, 2014)

Mit dieser vereinfacht erklärten Funktionsweise wird Strom erzeugt.

2.1.2 Funktionsweise der Solaranlage

Abbildung 2.2: Funktionsweise der Solaranlage



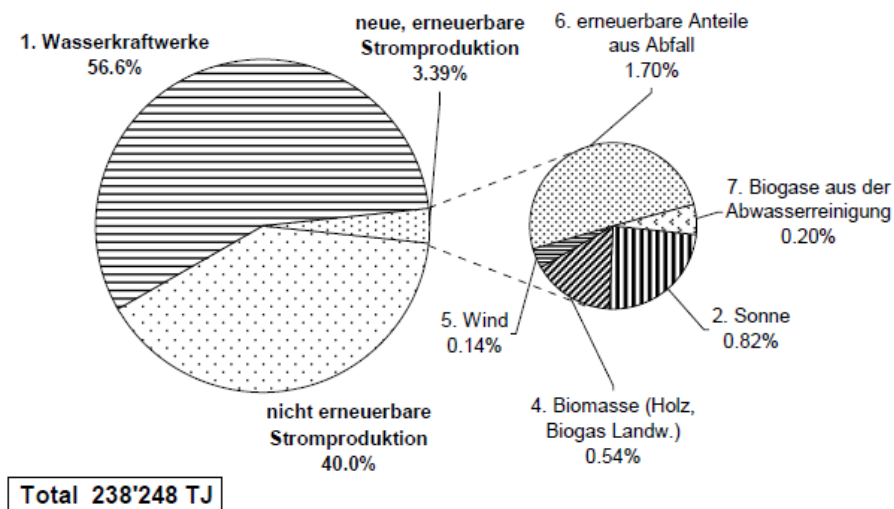
Quelle: (Centralschweizerische Kraftwerke AG, 2014)

In den Solarzellen wird, durch die von den Sonnenstrahlen ausgehende Energie, Gleichstrom erzeugt. Dieser Gleichstrom fließt dann zum Wechselrichter, welcher die Aufgabe hat, Gleichstrom in Wechselstrom umzuwandeln. Diese Umwandlung ist nötig, um schlussendlich den Strom ins Netz einzuspeisen. Der Wechselstrom fließt zudem auch durch einen Zähler, um zu sehen, wie viel Strom produziert wird. Der Strom kann nun vor Ort verbraucht werden. (Centralschweizerische Kraftwerke AG, 2014)

Produziert die Solaranlage an einem sonnigen Tag mehr Strom als verbraucht wird, dann fließt der überschüssige Strom ins öffentliche Netz („Energie-Rückspeisung ins Netz“). Wird umgekehrt bei schlechtem Wetter oder abends mehr Strom verbraucht als die Solaranlage produziert, fließt vom Netz der fehlende Strom ins Haus („Energie-Bezug ab Netz“). (Centralschweizerische Kraftwerke AG, 2014)

2.2 Schweizer Anteil an Solarenergie

Abbildung 2.3: In der Schweiz produzierte Energie



Quelle: (Dr. Eicher und Pauli AG, 2014)

In der oben abgebildeten Statistik ist ersichtlich, dass im Jahre 2014 lediglich 0.82% der in der Schweiz generierten Energie der Sonne zuzuschreiben ist (Dr. Eicher und Pauli AG, 2014).

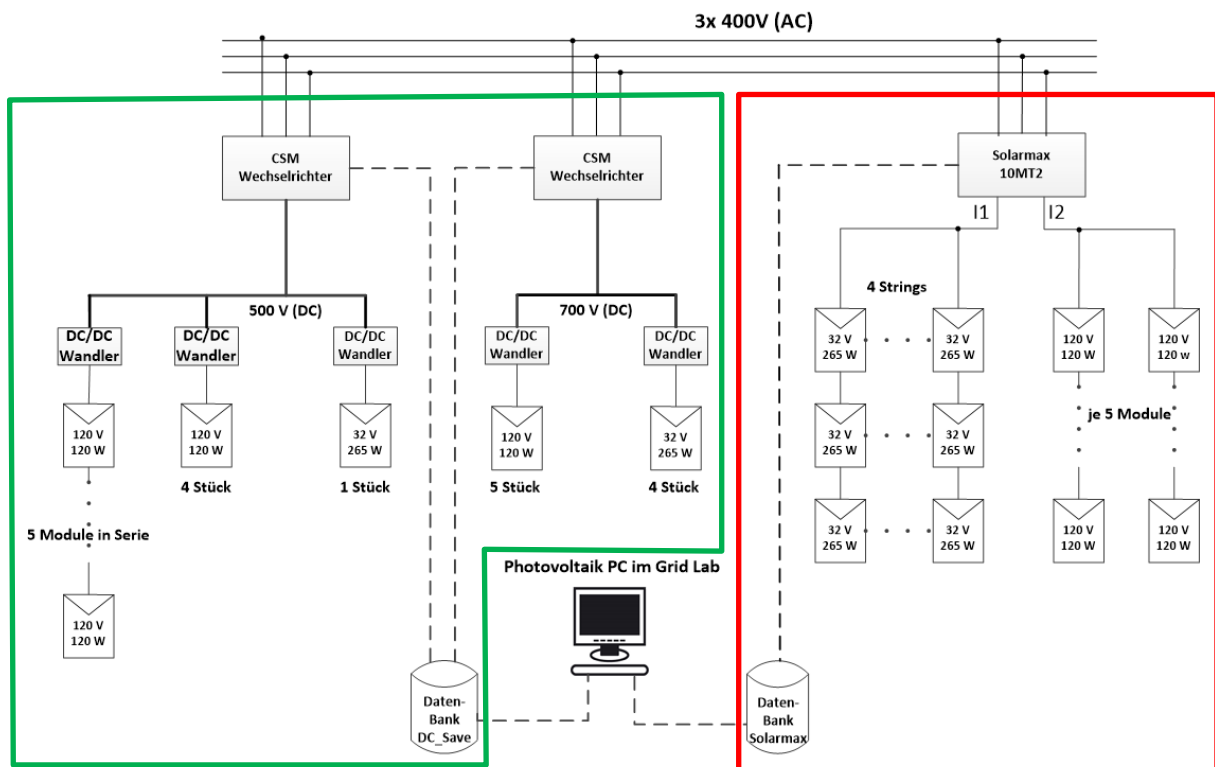
Im Jahre 2013 lag der Anteil der Solarstromproduktion noch bei 0.8%, was jedoch um die 70% mehr war als im Jahr zuvor. Im Vergleich wurde in Deutschland im Jahre 2012 lediglich 5.3% produziert. (Schweizerische Energie-Stiftung, 2014)

Gemäss Prognosen des Fachverbands Swissolar kann in der Schweiz die Solarstromproduktion auf bestehenden Dachflächen bis 2025 auf 20% erhöht werden. Dieser Wert entspricht der heutigen Jahresproduktion von vier AKW Mühleberg. Über einen längeren Zeitraum sind sogar bis zu 40% Solarstromanteil möglich. (Schweizerische Energie-Stiftung, 2014)

3. Porträt der PV-Anlage in Sion

In den letzten Jahren erlebten die Entwicklung sowie die Nachfrage nach Solarenergie einen starken Aufschwung. Daher entschied sich die HES-SO in Sion, eine eigene PV-Anlage auf ihrem Gebäude zu installieren. In der nachfolgenden Abbildung 3.1 ist diese installierte PV-Anlage schematisch dargestellt, welche aus dem Solarmax sowie einem 500 und einem 700 Volt DC-Bus besteht. Die aus den drei Systemen generierte Energie wird schliesslich ins Netz gespeist.

Abbildung 3.1: Übersicht der PV-Anlagen der HES-SO in Sion



Quelle: Eigene Darstellung in Zusammenarbeit mit Ivan Bittel

Der in der Abbildung 3.1 rot markierte Bereich (Solarmax) wird in dieser Bachelorarbeit implementiert. Der grüne Bereich (500V und 700V DC-Bus) wird in der Bachelorarbeit von Ivan Bittel eingeführt. Schlussendlich werden die Daten beider Anlagen auf dem Photovoltaik-PC im Büro „GridLab“ in Sion grafisch dargestellt.

4. Recherchephase

Zu Beginn wurde ein persönliches Gespräch mit dem Auftraggeber, Herrn Christoph Elert, durchgeführt. Es ging darum, spezifische Informationen zum Auftrag zu erhalten.

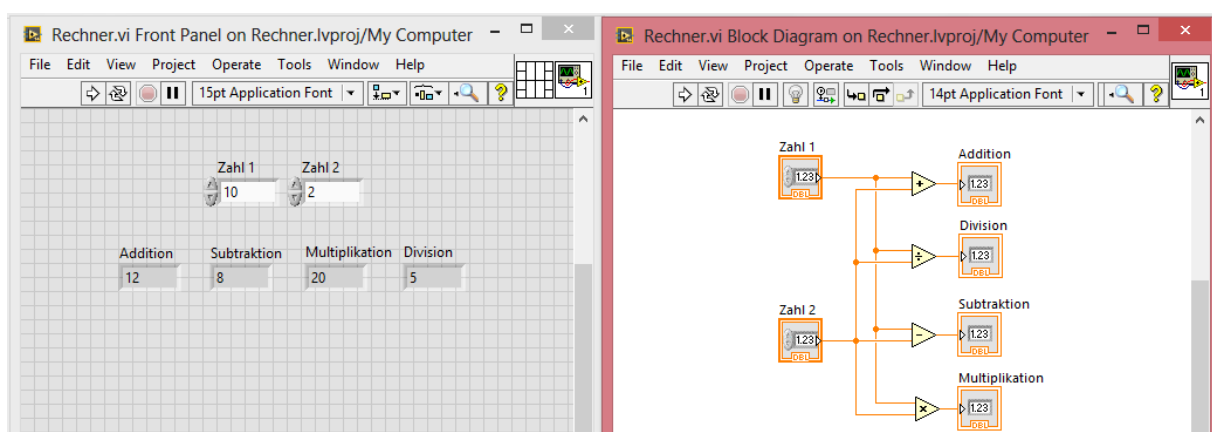
Für die Entwicklung des Systems wurde das Programm LabView zur Verfügung gestellt. Da im Studium nie graphisch programmiert wurde, ging es primär um das Kennenlernen des Programmes, was mithilfe verschiedener Online Tutorials, Videos und Büchern erreicht werden konnte.

4.1 LabView 2014

Die Entwicklungsumgebung LabView ist eigens zur Steigerung der Produktivität von Ingenieuren und Wissenschaftler ausgelegt worden. LabView ist ein wertvolles Werkzeug, um mit seiner grafischen Programmiersyntax technische Systeme visualisieren, erstellen und kodieren zu können. Ingenieure werden bei ihrer Ideenumsetzung unterstützt, Prüfzeiten werden reduziert und Erkenntnisse auf Grundlage gesammelter Daten geliefert. LabView ist seit Jahrzehnten die bevorzugte Lösung zum Entwickeln, Verteilen und Testen von Komponenten, vom Erstellen intelligenter Maschinen bis hin zur Qualitätssicherung vernetzter Geräte. (National Instruments Corporation)

Um das Funktionsprinzip von LabView zu veranschaulichen, wurde ein kleines Beispielprogramm erstellt:

Abbildung 4.1: Beispielprogramm LabView Rechner



Das Fenster auf der linken Seite ist der sogenannte Front Panel, also das User Interface. Die Elemente „Zahl 1“ und „Zahl 2“ sind Controls bzw. Eingaben, in welchen benutzerdefinierte Zahlen eingegeben werden können. Die Elemente „Addition“, „Subtraktion“, „Multiplikation“ und „Division“ sind die Indikatoren, also die automatische Ausgaben des Programmes. Das Fenster auf der rechten Seite ist das Block Diagram, die Logik hinter dem Front Panel. Hier werden „Zahl 1“ und „Zahl 2“ jeweils addiert, dividiert, subtrahiert und multipliziert.

Wird das Programm ausgeführt, werden die Zahlen, welche in den Feldern von „Zahl 1“ und „Zahl 2“ eingetragen werden, im Block Diagram verarbeitet und im Front Panel in den Indikatoren ausgegeben.

4.2 Photovoltaik

Da auch Photovoltaik ein für den Verfasser unbekanntes Gebiet ist, galt es auch dort zu recherchieren. Erste allgemeine Informationen gab es von Herrn Christoph Ellert, weitere und spezifischere Informationen konnten über verschiedene Informationskanäle eingeholt werden. Da im theoretischen Hintergrund (Punkt 2) das Gebiet bereits erläutert wurde, wird es hier nicht mehr erklärt.

5. Analysephase

Bis anhin existierte bereits eine Software, welche jedoch nicht umgesetzt werden konnte, da verschiedene Anforderungen nicht exakt erfüllt wurden. Dieses Programm wurde daraufhin von Ivan Bittel in einer Projektarbeit angepasst.

Das angepasste Programm konnte danach genutzt werden, um die Handhabung mit LabView besser zu verstehen und zu erlernen.

Die Grundfunktionen von LabView sind in den Tutorials ziemlich klar und einfach beschrieben. Doch gerade bei diesem Programm gibt es viele Unterfunktionen, welche einen sehr komplexen Aufbau aufzeigen. Der grösste Lerneffekt bestand darum beim Learning-by-doing.

6. Implementierungsphase

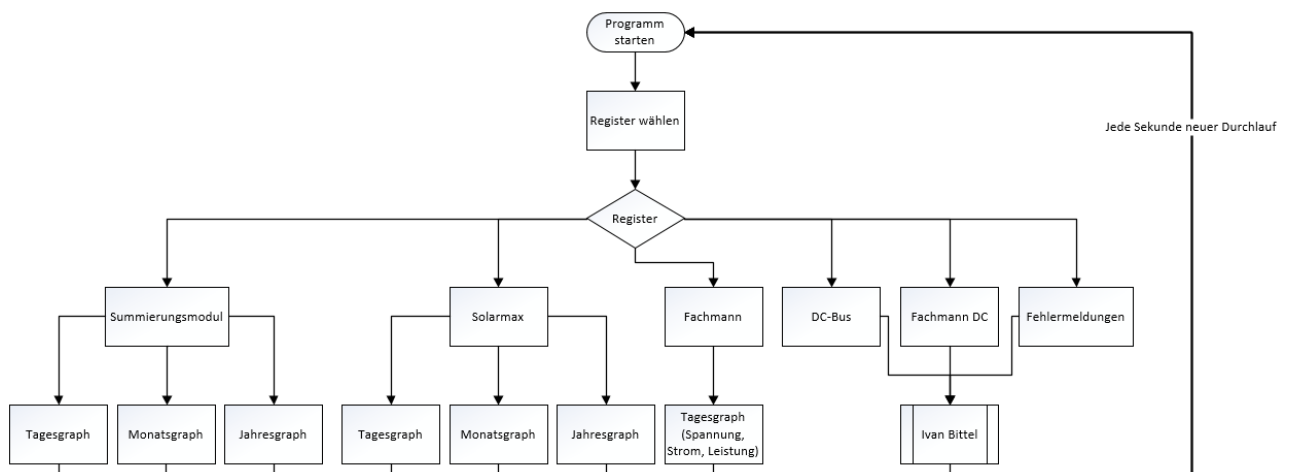
Zu Beginn dieser Phase musste die Software LabView 2014 auf dem privaten Computer installiert werden. Die Software wie auch der Lizenzschlüssel wurde von Herrn Aldo Vaccari zur Verfügung gestellt. Nach den ersten Tutorials und den Analysen am bestehenden Programm ging es darum, die Software zu entwickeln. Die Software umfasst folgende Bereiche:

Abbildung 6.1: Register der Software



Anhand des Flussdiagramms wird nun die Struktur der Software einfach dargestellt.

Abbildung 6.2: Flussdiagramm Programmablauf



Quelle: Eigene Darstellung mithilfe von Microsoft Visio

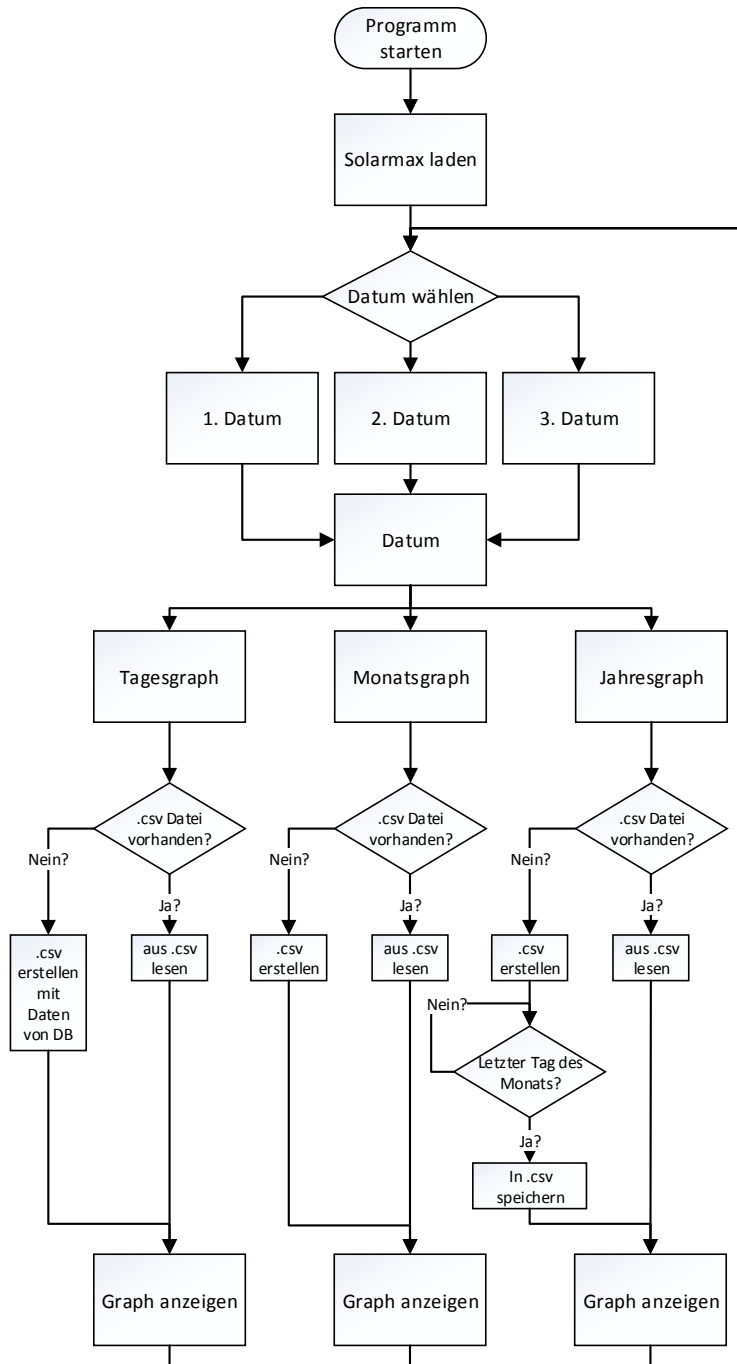
In dem oben aufgezeigten Flussdiagramm wird die Funktionsweise des Programmes dargestellt. Beim Starten des Programmes hat man die Möglichkeit, zwischen verschiedenen Registern zu wählen. Bei jedem der gewählten Register erscheinen verschiedene Graphen. Beim Summierungsmodul wie auch beim Solarmax gibt es einen Tagesgraphen, Monatsgraphen wie auch einen Jahresgraphen. Beim Register Fachmann wird nur eine Tageskurve mit Strom, Spannung und Leistung angezeigt. Die Register DC-Bus, Fachmann DC und Fehlermeldungen werden von Ivan Bittel implementiert und in dieser Arbeit darum hier nicht weiter berücksichtigt.

Um die Funktionen der Register besser zu verstehen, werden diese nachfolgend einzeln aufgeführt und erklärt.

6.1 Register Solarmax

In diesem Register können Daten von einem bestimmten Datum aus der DB der Solaranlage geladen, in eine .csv-Datei gespeichert und anschliessend in einem Graphen angezeigt werden.

Abbildung 6.3: Flussdiagramm Solarmax



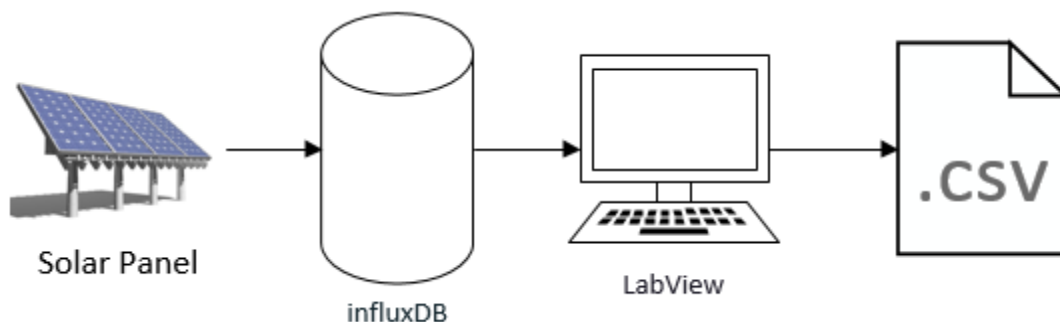
Quelle: Eigene Darstellung mithilfe von Microsoft Visio

In diesem Flussdiagramm wird die Funktionsweise des Solarmax dargestellt. Wird das Programm gestartet, wird der aktuelle Tag geladen, was hier als „1. Datum“ angezeigt wird. Zusätzlich können zwei frei wählbare Tage („2.Datum“ und „3.Datum“) gewählt werden. Von diesem Datum wird bei jedem einzelnen Graphen eine .csv-Datei erstellt, falls noch keine vorhanden ist. Danach werden die Daten aus der Datei ausgelesen und in den Graphen (Tages-, Monats- und Jahresgraphen) dargestellt.

6.1.1 Verbindungsaufbau zur Datenbank influxDB

Folgend wird der Verbindungsaufbau zur DB der Solaranlage influxDB aufgezeigt. Im Gegensatz zu anderen Programmausschnitten wird dieser im Detail erklärt, da die Verbindung zur DB essenziell ist.

Abbildung 6.4: vom Solarpanel zur .csv-Datei



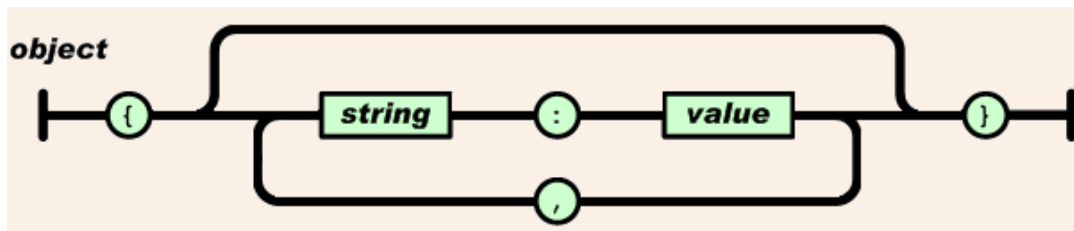
Quelle: Eigene Darstellung mithilfe von Microsoft Visio

Die Informationen der Solar Panels (Leistung, Strom, Spannung, usw.) werden in die Datenbank influxDB gespeichert. Anschliessend werden die Daten über das LabView Programm geladen, verarbeitet und in eine .csv-Datei geschrieben. Der detaillierte Prozess wird in diesem Kapitel beschrieben.

Nach einigem Probieren war es möglich, den Verbindungsaufbau zur Datenbank zu erstellen und die Daten in einem JSON-String auszugeben.

Bei JSON (JavaScript Object Notation) handelt es sich um ein Datenaustauschformat, welches für den Menschen einfach zu lesen und zu schreiben ist. Für Maschinen ist es leicht zu parsen und zu generieren. Es handelt sich bei JSON um ein Textformat, welches unabhängig von Programmiersprachen ist, aber vielen Konventionen folgt. Darum ist JSON ein ideales Format für den Datenaustausch. (Einführung in JSON)

Abbildung 6.5: Darstellung JSON Objekt

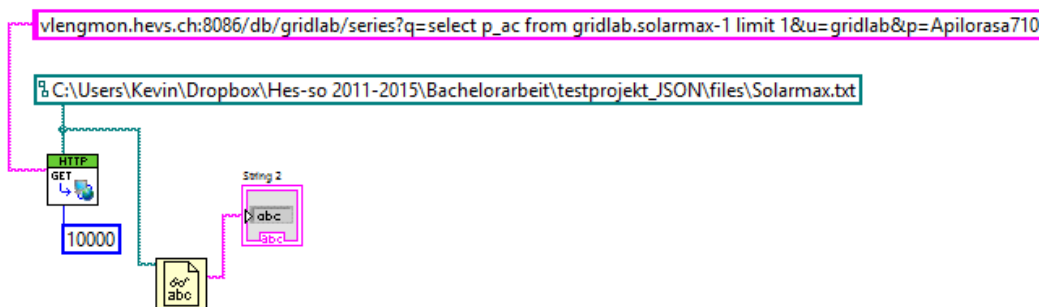


Quelle: (Einführung in JSON)

„Ein Objekt ist eine ungeordnete Menge von Name/Wert Paaren. Ein Objekt beginnt mit { (geschwungene Klammer auf) und endet mit } (geschwungene Klammer zu). Jedem Namen folgt ein : (Doppelpunkt) gefolgt vom Wert und die einzelnen Name/Wert Paare werden durch , (Komma) voneinander getrennt“ (Einführung in JSON).

Zu Beginn wurde die Verbindung anhand dieses kleinen Beispiels getestet, welches nur den zuletzt hinzugefügten Eintrag der Datenbank ausgibt. Aus der Datenbank liest es den Parameter p_ac aus, also die wechselstromseitige Gesamtleistung.

Abbildung 6.6: Aufbau und Test der Datenbankverbindung



Nachstehend werden die einzelnen Elemente beschrieben:

1)


vlengmon.hevs.ch:8086/db/gridlab/series?q=select p_ac from gridlab.solarmax-1 limit 1&u=gridlab&p=Apilorasa710

Mit diesem Element wird die Datenbankverbindung erstellt. Es richtet sich nach folgender Struktur:

Tabelle 6.1: Beschreibung der Elemente für die Datenbankverbindung

Element	Beschreibung
vlengmon.hevs.ch:8086	Host Name mit Port
/db/	Fixes Element
gridlab	Name der Datenbank
/series?q=	Fixes Element für SQL Abfrage
select p_ac from gridlab.solarmax-1 limit 1	SQL Abfrage, um die gewünschten Werte aus der Datenbank zu erhalten.
&u=	Fixes Element für Benutzernamen
gridlab	Benutzername für die Datenbank
&p=	Fixes Element für Passwort
Apilorasa710	Passwort für die Datenbank

2)

 C:\Users\Kevin\Dropbox\Hes-so 2011-2015\Bachelorarbeit\testprojekt_JSON\files\Solarmax.txt

Die von der Datenbank geladenen Daten werden in diesem Verzeichnis in die Solarmax.txt-Datei abgespeichert.

3)



Mit diesem http GET werden die Daten aus der Datenbank geladen und in die Datei geschrieben.

4)

 10000

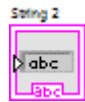
Mit diesem timeout von 10'000 Millisekunden oder 10 Sekunden wird erreicht, dass wenn nach 10 Sekunden keine Daten geladen werden konnten, der Prozess abgebrochen wird.

5)



Hiermit können die Daten aus der Datei Solarmax.txt gelesen werden. Diese werden jedoch 1:1 in dem JSON Format angezeigt, wie sie geladen wurden. Es ist also notwendig, dass die Daten in ein angenehmes Format konvertiert werden, was in Abschnitt 6.1.3 erläutert wird.

6)

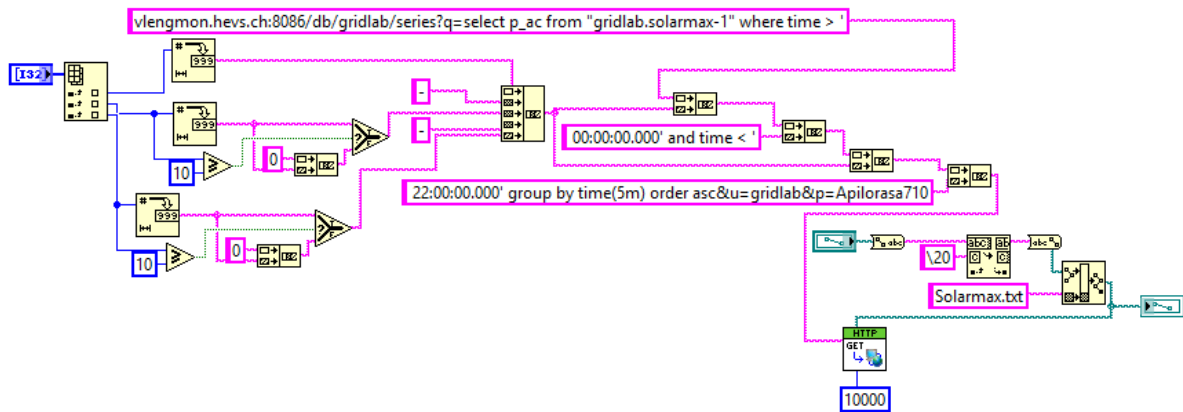


Schlussendlich werden die geladenen Daten im JSON Format in einem Textfeld ausgegeben.

6.1.2 Daten von beliebigem Datum laden

In einem weiteren Schritt geht es darum, die Tagesdaten eines gewählten Datums aus der Datenbank zu lesen. Dafür muss der Prozess, welcher in Abbildung 6.6 beschrieben wurde, wie folgt ergänzt werden:

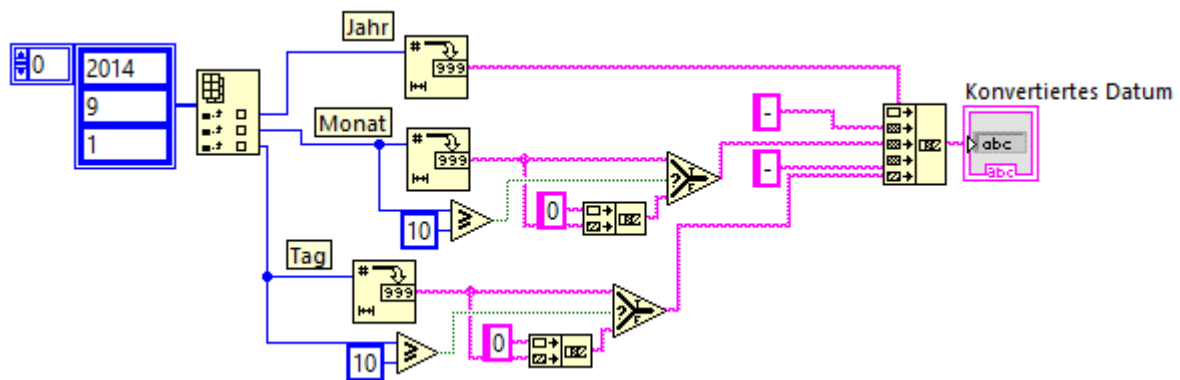
Abbildung 6.7: Ergänzte Datenbankabfrage mit gewähltem Datum



Damit die SQL-Abfrage dynamisch erfolgt und demzufolge immer vom entsprechend gewählten Datum die DB-Einträge geladen werden, wird das Datum direkt in die SQL-Abfrage eingebunden. Wenn z.B. das Datum 16. Juli 2015 gewählt wird, sieht die DB-Verbindung folgendermassen aus: `vlengmon.hevs.ch:8086/db/gridlab/series?q=select p_ac from „gridlab.solarmax-1“ where time>'2015-07-16 00:00:00.000 and time<'2015-07-16' group by time(5m) order asc&u=gridlab&p=Apilorasa710`

Bei einem Datum, in welchem der Tag und der Monat kleiner als 10 sind (z.B. 01.09.2014), bestand das Problem darin, dass das Datum als Zahl weitergegeben wird. Die 0 wird jeweils ignoriert, wodurch 1.9.2014 ausgegeben wird. Diese Ausgabe wird aber nicht als Datum akzeptiert. Folglich musste hier eine Lösung gefunden werden. Darum wurden entsprechende Anpassungen vorgenommen. Die Lösung sieht folgendermassen aus:

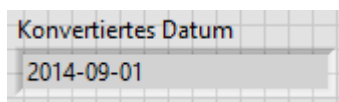
Abbildung 6.8: Konvertierung von Datum 1.9.2014 nach 01.09.2014



In diesem Prozess wird zuerst ein Datum erhalten, welches im Programm ausgewählt wurde. In diesem Fall ist das der 1.9.2014. In einem nächsten Schritt wird dieses Datum in die einzelnen Bestandteile zerlegt, also in Jahr, Monat und Tag. Das Jahr kann ohne eine weitere Verarbeitung weitergegeben werden. Beim Monat wie auch beim Tag wird die Überprüfung gemacht, ob die Zahl kleiner ist als 10. Ist dies der Fall, wird vor die Zahl eine 0 geschrieben. Zum Schluss wird das Datum wieder zusammengefügt.

Nach der Bearbeitung wird das Datum schliesslich im richtigen Format ausgegeben, um es weiter zu verwenden, wie in der folgenden Abbildung zu sehen ist:

Abbildung 6.9: Resultat Konvertierung von Datum 1.9.2014 nach 01.09.2014



6.1.3 Verarbeitung vom JSON String und speichern in .csv-Datei

Werden die Daten von der Datenbank heruntergeladen und in die Solarmax.txt-Datei gespeichert, bestehen diese in einem JSON Format. Beim JSON Format werden die Daten ohne Zeilenumbruch aneinander geschrieben. Durch die eckigen Klammern [] wird gekennzeichnet, welche Zahl zu welchem Eintrag der Datenbank gehört.

Darum geht es in einem nächsten Schritt darum, die heruntergeladenen Daten weiterzuverarbeiten, um sie dann in einem geeigneten Format in eine .csv-Datei zu speichern. Später werden diese dann vom Programm automatisch aus der Datei gelesen, um sie grafisch darzustellen.

Wenn die Daten von der Datenbank in die Datei Solarmax.txt gespeichert werden, sieht das folgendermassen aus:

Abbildung 6.10: JSON String aus Datenbank

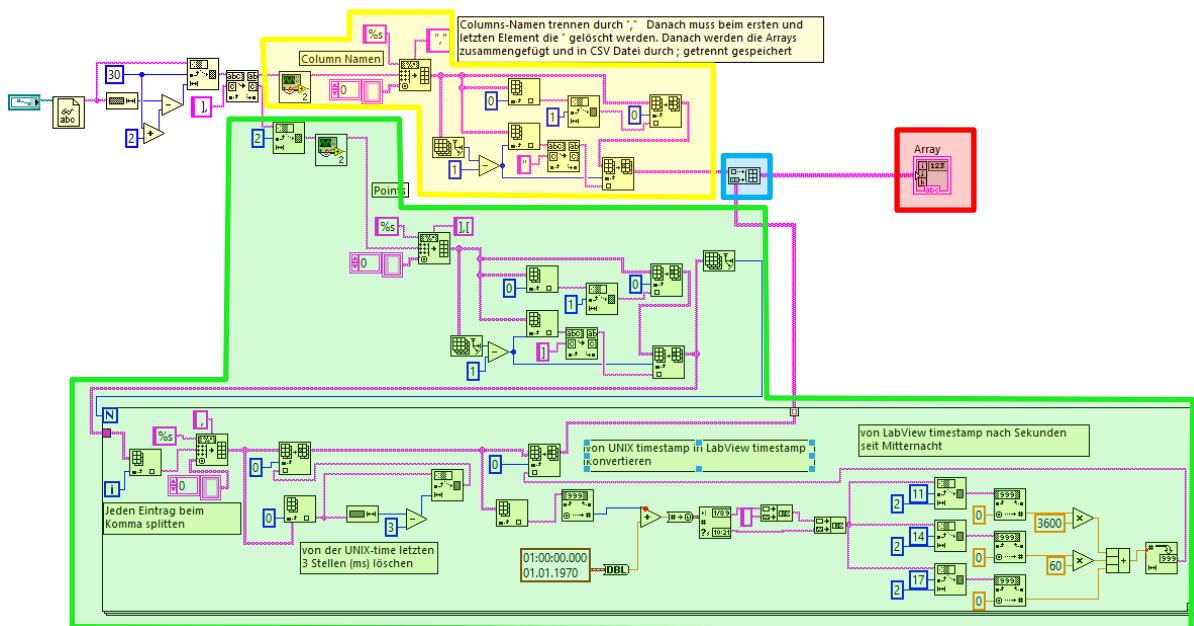
```
JSON String
[[{"name":"gridlab.solarmax-1","columns":["time","sequence_number","p_ac"],"points":[[1432974990142,1906570001,1008],
[1432974980142,1906480001,1012],[1432974970143,1906390001,1012],[1432974960142,1906300001,1009],[1432974950143,
1906210001,1007],[1432974940142,1906110001,1006],[1432974930142,1906010001,1004],[1432974920142,1905920001,1006],
[1432974910142,1905830001,1004],[1432974900143,1905740001,1005]]]}
```

Mit „columns“ wird angegeben, welche Tabellen aus der Datenbank geladen werden. Die Tabellen mit den Namen „time“, „sequence_number“ und „p_ac“ werden wie folgt im Code angegeben: (time, sequence_number, p_ac). Jede Tabelle beinhaltet 10 Einträge. Bei „points“ wird dementsprechend von jedem der 10 geladenen Einträge die columns angegeben. Für den ersten Eintrag [1432974990142,1906570001,1008] bedeutet dies:

- time = 1432974990142 (dieser Teil wird in der nächsten Abbildung im grünen Bereich genauer erläutert)
- sequence_number = 1906570001
- p_ac = 1008

Anschliessend wird der JSON String verarbeitet. Der Prozess sieht wie folgt aus:

Abbildung 6.11: Verarbeitung JSON String



Damit nicht jedes Element separat erklärt werden muss, werden nur die Funktionen der vier markierten Bereiche erklärt:

- **Gelb:** Hier wird die oberste Zeile (Header) aus den Zeilennamen für die .csv-Datei erstellt.
- **Grün:** Die Daten zu den entsprechenden Zeilennamen werden in diesem Bereich verarbeitet. Der Zeitstempel (time) wird im UNIX-Format gespeichert.

1969 wurde die Unixzeit als einfache Zeitdefinition für das Betriebssystem UNIX entwickelt. Die Unixzeit zählt die vergangenen Sekunden seit dem Donnerstag, den 01. Januar 1970 00:00 Uhr UTC. (Söbbing, 2014)

Abbildung 6.12: Eintrag mit UNIX Zeitstempel

```
time;sequence_number;p_ac
1432974990142;1906570001;1008
```

Diese musste in einen LabView Zeitstempel konvertiert werden (01.01.2015 10:00:00.000).

Abbildung 6.13: Eintrag mit LabView Zeitstempel

```
time;sequence_number;p_ac
05.06.2015 06:54;1906570001;1008
```

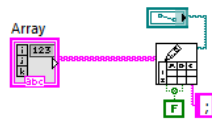
Danach wurde der LabView Zeitstempel wiederum in ein passendes Format konvertiert, damit die Daten vom Graphen eingelesen werden können (Sekunden, die seit Mitternacht vergangen sind. Also z.B. für 10:00 Uhr wäre dies der Wert 36'000).

Abbildung 6.14: Eintrag mit Zeitstempel Sekunden seit Mitternacht

```
time;sequence_number;p_ac  
24840;1906570001;1008
```

- **Blau:** Die zwei erstellten 1D Arrays (Bereich grün und gelb) werden in ein 2D Array gespeichert, damit jeder Eintrag im Header, z.B. time, sequence_number und p_ac x-beliebige Einträge haben kann. Zuerst wird das 1D Array vom gelben Bereich hinzugefügt, damit dieses in die erste Zeile geschrieben wird. Danach werden die jeweiligen Daten aus dem grünen Bereich hinzugefügt.
- **Rot:** Das 2D Array wird schlussendlich in einer weiteren Funktion in eine .csv-Datei geschrieben :

Abbildung 6.15: Daten in .csv-Datei speichern



Die Einträge werden durch ein Semikolon (;) getrennt.

Damit man die getrennten Einträge mit dem Semikolon erkennen kann, muss man die Datei mit dem Editor öffnen. Dies sieht dann folgendermassen aus:

Abbildung 6.16: Verarbeiteter JSON String in Solarmax.csv mit Editor

```
time;sequence_number;p_ac  
24840;959570001;201
```

Zur besseren Ansicht wird die Datei Solarmax.csv mit Microsoft Excel geöffnet:

Abbildung 6.17: Verarbeiteter JSON String in Solarmax.csv mit Microsoft Excel

	A	B	C
1	time	sequence_number	p_ac
2	24840	959570001	201

6.1.4 Daten aus .csv-Datei lesen und in einem Graphen anzeigen

Die Daten sind nun im richtigen Format in die .csv-Datei gespeichert. Es geht in einem weiteren Schritt darum, die Daten vom entsprechenden Tag in einem Graphen anzeigen zu lassen. Um dies zu erreichen, müssen einige Funktionen des bestehenden Programmes angepasst werden. Beim DC-Bus muss die Leistung zuerst mit der Formel $P=U \cdot I$ berechnet werden, beim Solarmax wird direkt die Leistung P in die Datenbank geladen.

6.1.4.1 Überblick des Programmes

Untenstehende Abbildung 6.18 zeigt das UI vom Überwachungsprogramm des Solarmax, was das Kernstück der Arbeit bildet.

Abbildung 6.18: Überblick des gesamten Programmes



- **Rot:** In der linken oberen Ecke ist das Schema des Solarmax abgebildet. Es zeigt an, wie die einzelnen Panels angeordnet bzw. miteinander verbunden sind sowie die Gesamtanzahl der Panels.
- **Grün:** Unten links wird die Tageskurve angezeigt, welche die produzierte Energie des laufenden Tages und von zwei weiteren frei wählbaren Tagen anzeigt. Zusätzlich kann jeweils zu diesen Tagen die Meteokurve angezeigt werden, welche die theoretisch mögliche Energiegewinnung ausgibt.

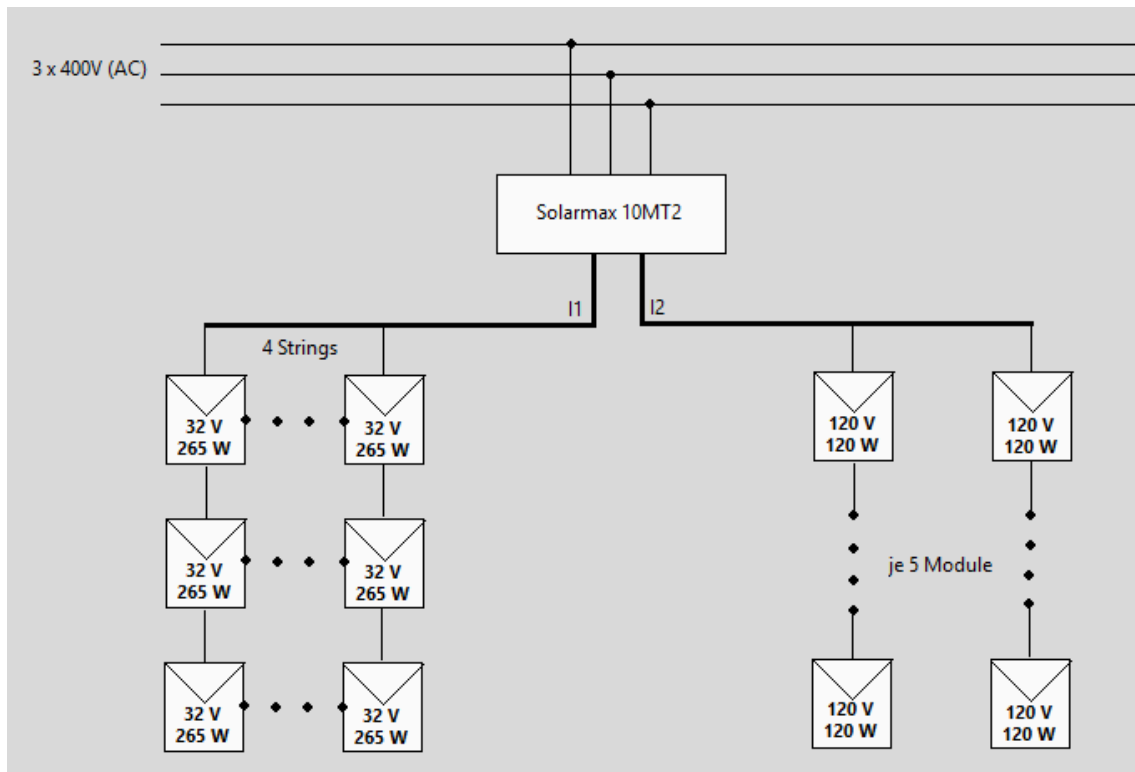
- **Blau:** Unten rechts wird die Monatskurve abgebildet. Auch bei der Monatskurve bietet sich die Möglichkeit, zwei weitere frei wählbare Monate neben dem aktuellen Monat anzeigen zu lassen. Es kann aber keine Meteokurve miteinbezogen werden. Da die erzeugte Energie des aktuellen Tages jede Minute automatisch aktualisiert wird, wird auch die Monatskurve dementsprechend jede Minute aktualisiert.
- **Gelb:** Schlussendlich wird der Jahresgraph in der rechten Ecke oben dargestellt. Im Gegensatz zu den beiden anderen Graphen wird dieser nicht laufend, sondern immer Ende des Monats aktualisiert. Jeweils am letzten Tag des Monats um 23.30 Uhr wird der Jahresgraph mit dem ablaufenden Monat ergänzt. Die schwarz gestrichelte Linie in diesem Graphen zeigt die Simulationskurve, welche unter Berücksichtigung der geographischen Lage die potenzielle Energiegewinnung anzeigt. Die Simulationskurve wird in Punkt 6.1.4.7 näher beschrieben.

Bei der Auswahl eines Datums wird immer zuerst überprüft, ob für dieses Datum bereits eine .csv-Datei mit den geladenen Daten existiert. Ist dies nicht der Fall, werden die Daten von der Datenbank in eine .csv-Datei geschrieben. Falls man also beispielsweise die Kurve vom 16.07.2015 anzeigen lassen möchte, muss dieses Datum in einem ersten Schritt ausgewählt werden. Anschliessend wird überprüft, ob von diesem Datum bereits eine Datei existiert. Es wird folglich im System nach dem folgenden Dateinamen gesucht: 2015_07_16_Solarmax.csv. Ist dies nicht der Fall, wird eine neue Datei für den betreffenden Tag erstellt. Beim nächsten Mal, wenn dieser Tag ausgewählt wird, muss die Datei nicht mehr neu erstellt werden. Die Daten werden direkt in den Graphen geladen.

In den nächsten sechs Kapiteln werden die einzelnen Graphen und deren Funktionsweisen näher erläutert.

6.1.4.2 Das Schema Solarmax

Abbildung 6.19: Schema Solarmax

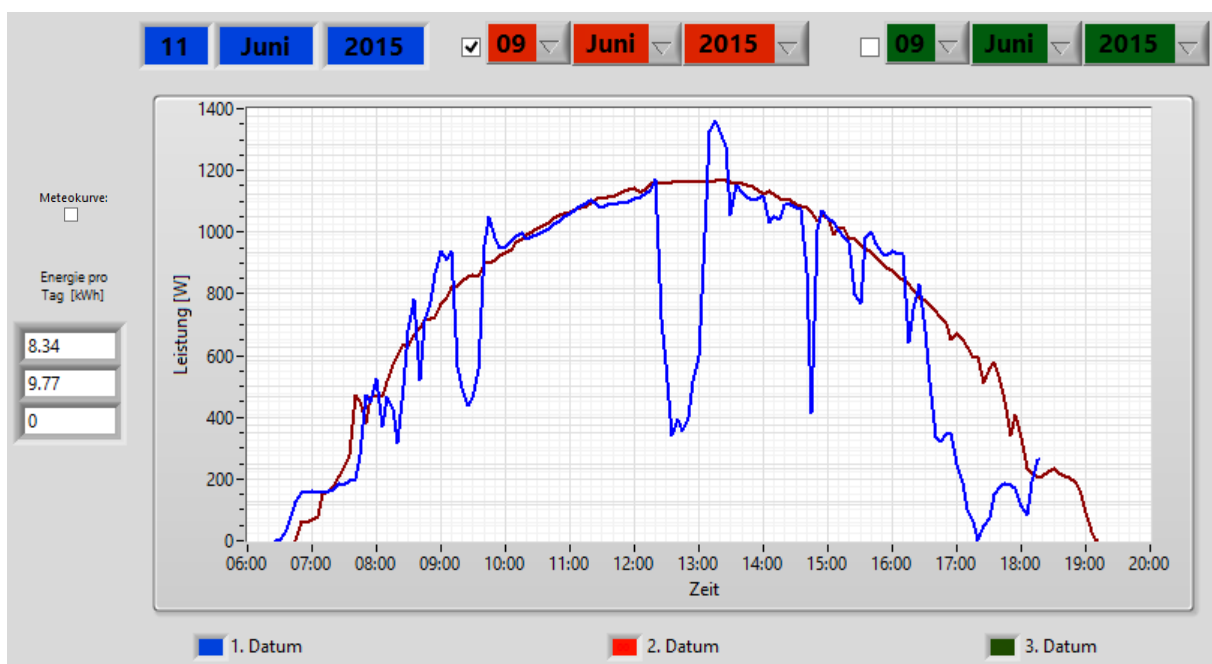


Beim Solarmax gibt es zwei Eingänge (Inputs): I1 und I2. Am I1 sind 12 Panels des Typs CSun angehängt. Hier sind jeweils drei Panels an vier Strängen (Strings) verbunden. Am I2 sind 10 Panels des Typs Inventux zu jeweils fünf Panels an zwei Strings angereicht.

6.1.4.3 Die Tageskurve

Beim Starten des Programmes werden die bisherigen Daten vom aktuellen Tag geladen und angezeigt. Links neben dem Graphen wird die gesamtproduzierte Energie des entsprechenden Tages ausgegeben. Die blaue Kurve visualisiert die Werte des aktuellen Tages. Oberhalb des Graphen können optional zwei gewünschte Daten angewählt werden (roter und grüner Bereich). Will man diese Daten im Graphen anzeigen lassen, müssen dafür die beiden Kontrollkästchen neben dem Datum aktiviert werden. Es besteht auch die Möglichkeit, nur ein Datum zusätzlich zum aktuellen Tag anzeigen zu lassen. In dieser Grafik ist nur das rote Datum aktiviert.

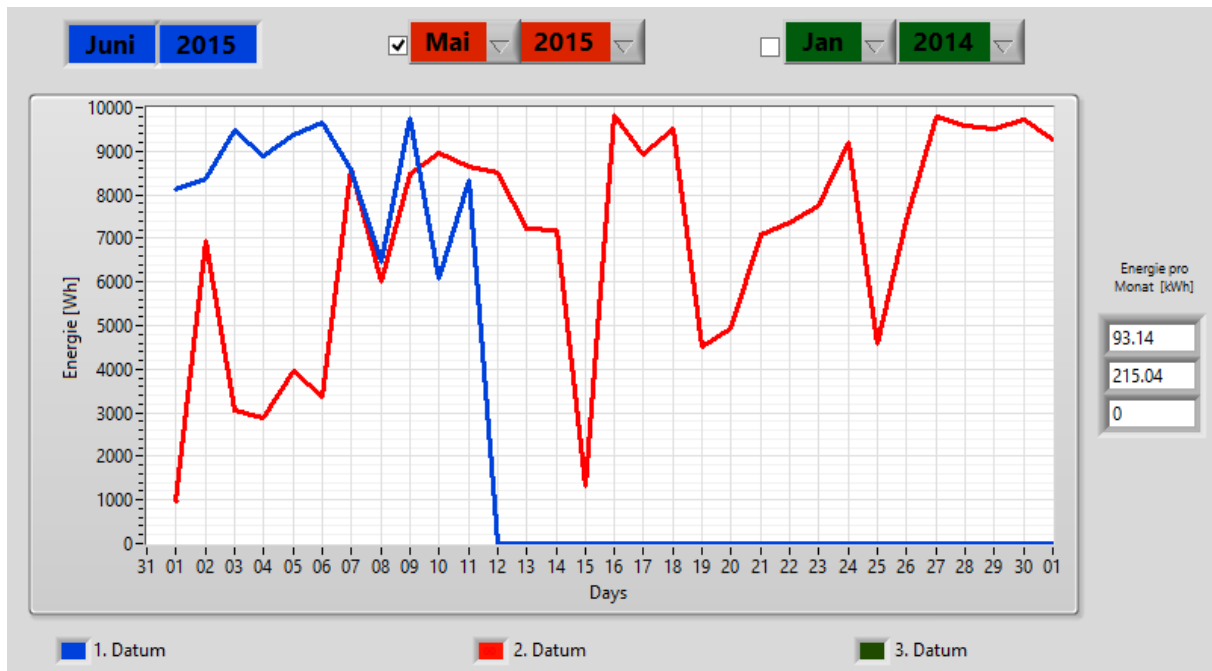
Abbildung 6.20: Die Tageskurve



In der obigen Abbildung ist sehr gut ersichtlich, dass am 09. Juni 2015 die Energiegewinnung hoch war. Darum lässt sich ableiten, dass an diesem Tag klares, wolkenloses Wetter war. Im Gegensatz ist ersichtlich, dass am 11. Juni 2015 zeitweise Wolken auftraten, so z.B. zwischen 12:00 Uhr und 13:00 Uhr wie auch kurz vor 15:00 Uhr. Während diesen Zeiten konnte weniger Energie generiert werden.

6.1.4.4 Die Monatskurve

Abbildung 6.21: Die Monatskurve



Aus dieser Grafik ist sehr gut abzulesen, wie das Wetter an den gewählten Tagen war. Bei der roten Linie (Mai 2015) sieht man sehr gut, dass es am 01. Mai und am 15. Mai stark bewölkt war und darum sehr wenig Energie produziert werden konnte.

6.1.4.5 Die Jahreskurve

Die Implementierung der dritten Grafik (Jahreskurve) ist zeitaufwendig. Bevor die Daten in einem Graphen dargestellt werden können, muss eine .csv-Datei erstellt werden (2015_00_00_Solarmax.csv). Diese Datei ist wie folgt aufgebaut:

Abbildung 6.22: Jahresdaten 2015_00_00_Solarmax.csv

3534361200	0
3531682800	0
3529004400	0
3526326000	0
3523647600	0
3520969200	0
3518290800	210246.305
3515612400	210082.485
3512934000	26907.913
3510255600	0
3507577200	0
3504898800	0

In dieser Tabelle wird die Energiegewinnung eines Jahres, eingeteilt in die zwölf Monate, dargestellt. In umgekehrter Reihenfolge werden die Monate aufgelistet, so also wird zuunterst das Total der generierten Energie vom Monat Januar gespeichert, zuoberst diejenige vom Dezember. Die Werte sind in kWh angegeben. Diese Daten werden anschliessend vom System ausgelesen und dargestellt. Jedoch stellte sich nach einem Test heraus, dass die falschen Werte ausgegeben werden, da sie nicht mit den Soll-Werten übereinstimmen. In der nachstehenden Abbildung sind die Werte von den Monaten Juni, Mai und April ersichtlich, welche eigentlich in die .csv-Datei gespeichert werden sollen:

Abbildung 6.23: Soll-Werte Juni, Mai & April

Energie pro Monat [kWh]
181.74
215.04
51.42

Nach Überprüfung des Fehlers kam man auf folgende Schlussfolgerung:

- Der Wert für den Monat April (26'907.913) wird nur vom 01. April bis zum 28. April zusammenaddiert.

- Der Wert für den Mai (210'082.485) ergibt sich aus der Summe vom 28. April-30. April und 2. Mai bis 28. Mai.
- Beim Juni (210'246.305): 29. Mai bis 31. Mai und 1. Juni bis 28. Juni.

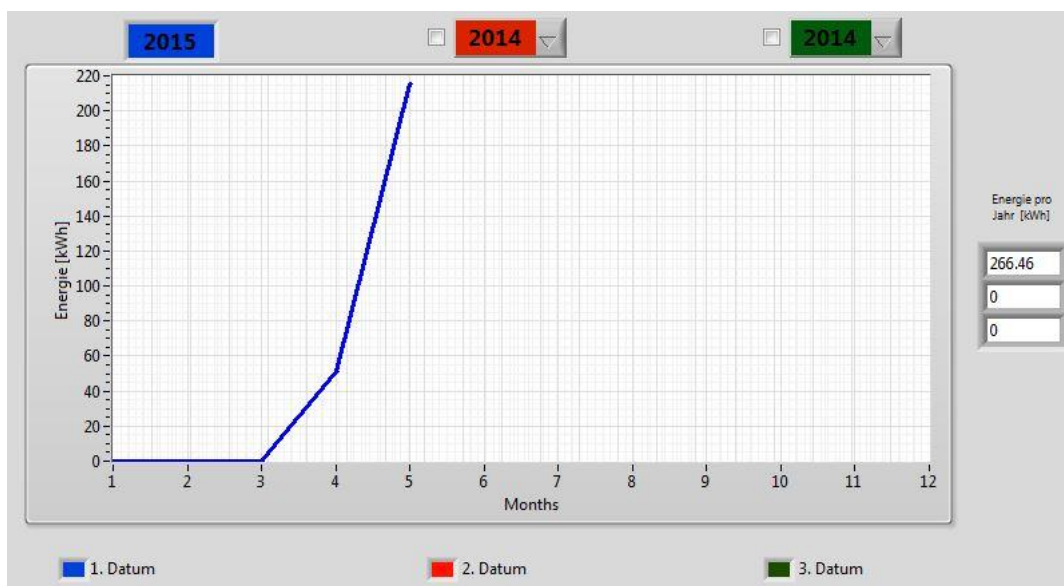
Um den konkreten Fehler zu finden, wurde das bestehende Programm analysiert. Es stellte sich heraus, dass die Jahreskurve auch im bestehenden Programm nicht richtig funktioniert. Bereits bei diesem Programm bestand das Problem, dass die Daten zwar ausgelesen und dargestellt, jedoch die Daten der Monate vermischt wurden. Darum musste eine Lösung gesucht werden, welche schliesslich daraus bestand, die Daten erst Ende Monat in die Jahresdatei 2015_00_00_Solarmax.csv schreiben zu lassen. Da z.B. am 15. Juni der Monat Juni noch nicht im Jahresgraphen dargestellt wird, kann man diesen nur im Monatsgraphen sehen.

Die Jahresdatei und die Jahreskurve sehen vor dem 01. Juli folgendermassen aus:

Abbildung 6.24: Jahresdaten bis Mai

1	0
2	0
3	0
4	51.42
5	215.04

Abbildung 6.25: Jahreskurve bis Mai

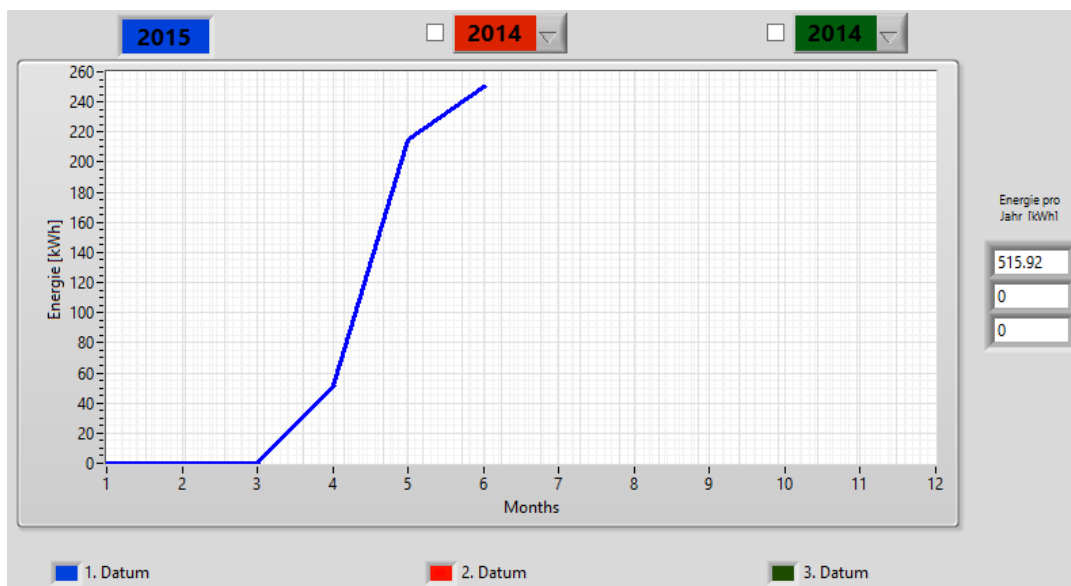


Wyer Kevin

Es ist ersichtlich, dass die Werte der vorgehenden Monate Januar bis Mai in die Datei aufgenommen wurden, aber noch keine Werte des Monats Juni. Der Monat Juni wird, wie bereits erwähnt, erst am Ende des Monats, d.h. am 30. Juni 2015 um 23.30 Uhr in die Datei geschrieben.

Nach erfolgreichem Laden der Daten des Monats Juni sieht die Jahreskurve am 01. Juli folgendermassen aus:

Abbildung 6.26: Jahreskurve am 01. Juli



In der Datei 2015_00_00_Solarmax.csv wird eine neue Zeile mit dem neuen Monat eingefügt:

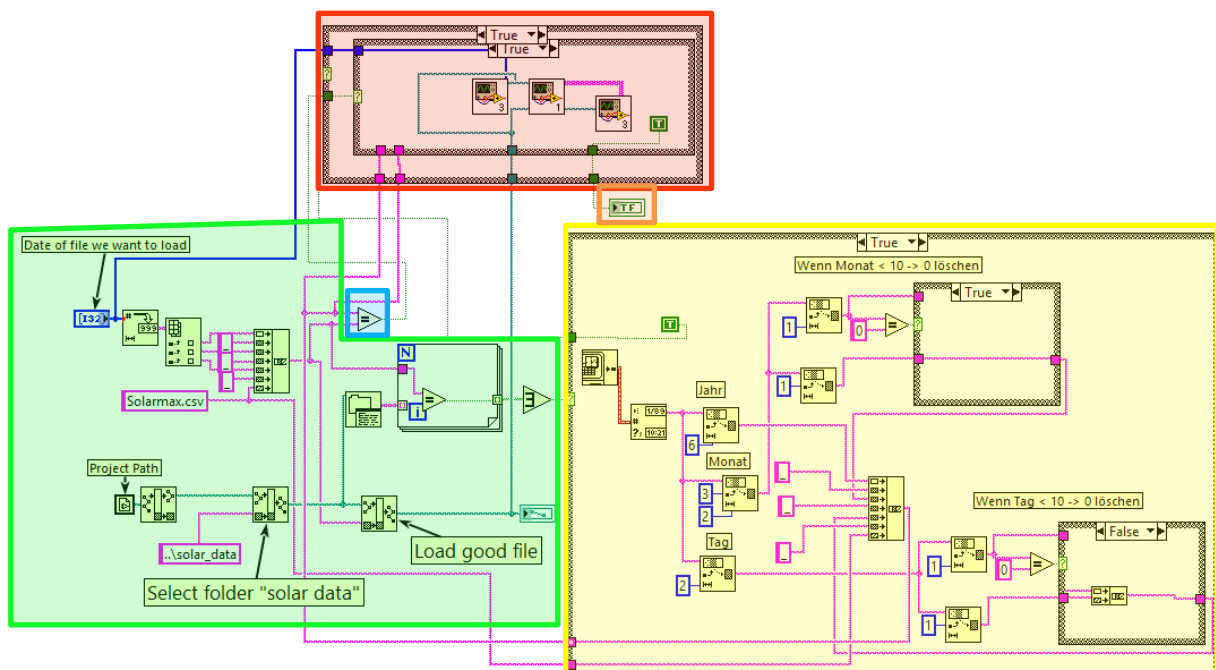
Abbildung 6.27: Jahresdaten am 01. Juli

1	0
2	0
3	0
4	51.36
5	214.87
6	249.69

6.1.4.6 Daten zur Laufzeit

Beim Laden der Daten eines Tages stiess man auf ein weiteres Problem. Die Daten eines Tages werden, sobald das Programm gestartet wird, automatisch in eine Datei gespeichert, um sie danach im Graphen anzuzeigen. Wenn also das Programm um 12:00 gestartet wird, werden die Werte bis um 12:00 Uhr aus der DB geladen und in eine .csv-Datei gespeichert. Die Datei wird aber nun nicht laufend automatisch aktualisiert, d.h. am Ende des Tages sind immer noch nur die Werte bis 12:00 Uhr in der Datei und somit zeigt der Graph auch nur diese Werte an, wodurch das Ergebnis verfälscht wird. Das Problem liegt also darin, dass bevor neue Daten von der DB geladen werden, überprüft wird, ob am gewählten Tag bereits eine Datei existiert. Falls dies der Fall ist, werden keine neuen Daten geladen, sondern die Daten der existierenden Datei direkt im Graphen angezeigt. Um dies zu umgehen, musste eine weitere Funktion erstellt werden:

Abbildung 6.28: Daten vom aktuellen Tag laufend aktualisieren



Hier wird überprüft, ob die geladenen Daten vom heutigen Tag sind. Ist dies der Fall, wird die Datei laufend aktualisiert.

Diese Funktion setzt sich wie folgt zusammen:

- **Grün:** In diesem Bereich wird überprüft, ob die Daten des ausgewählten Datums bereits in einer .csv-Datei existieren. Wenn nicht, wird für die Datei der Pfad erstellt. Dafür wird das gewählte Datum herausgenommen und eingebaut. Für z.B. den 10. Juni 2015 sieht der Pfad wie folgt aus: 2015_06_10_Solarmax.csv. Die Dateien werden in den Ordner „solar_data“ gespeichert. Sollte die Datei bereits existieren, werden die nächsten Schritte nicht weiter ausgeführt. Die Daten werden direkt aus der Datei gelesen und im Graphen angezeigt. Wenn die Datei vom 10. Juni 2015 bereits existiert, wird der Prozess im gelben Bereich ausgeführt.
- **Gelb:** Das aktuelle Datum wird aufgerufen und in das gleiche Format konvertiert wie im grünen Bereich: 2015_06_10_Solarmax.csv. Sollte der Monat oder der Tag kleiner als 10 sein, wird die 0 gelöscht (der Monat Juni 06 wird zu 6).
- **Blau:** Hier wird das aktuelle Datum und das ausgewählte Datum verglichen. Stimmen diese überein, durchläuft das Programm den roten Prozess. Wenn die beiden Daten nicht übereinstimmen, wird die bestehende Datei direkt ausgelesen und im Graphen dargestellt.
- **Rot:** In diesem Schritt werden dauerhaft die aktuellen Daten ausgelesen und in die .csv-Datei geschrieben. Damit wird erreicht, dass im Graphen die neuen Daten direkt angezeigt werden.
- **Orange:** Zum Schluss wird entweder True oder False weitergegeben.
 - **False:** Die Datei existiert bereits und kann direkt ausgelesen und grafisch dargestellt werden.
 - **True:** Die Datei existiert noch nicht und muss darum anhand des Pfades, welcher im grünen Bereich generiert wird, erstellt werden.

Nach der Implementierung dieses Prozesses wurden die Daten laufend, d.h. während der Ausführung des Programmes, geladen. In der Monats- wie auch in der Jahresgrafik werden die Daten somit auch laufend aktualisiert.

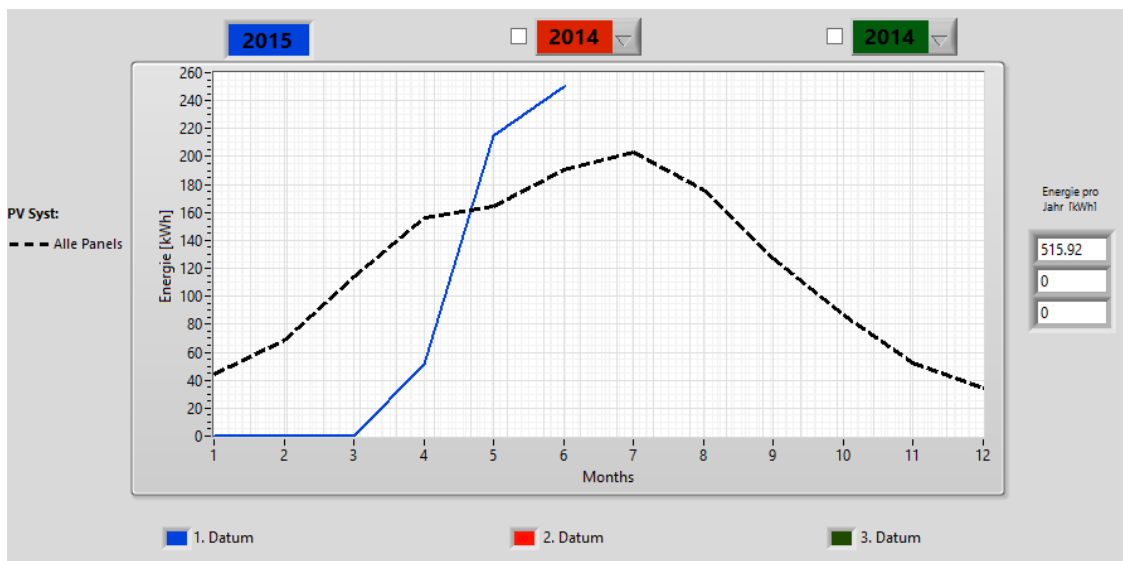
6.1.4.7 Simulationsdaten mit PVsyst berechnen und anzeigen

Die Berechnung der Simulationsdaten war Aufgabe von Ivan Bittel. In diesem Kapitel wird ein kurzer Ausschnitt aus seiner Arbeit aufgeführt, um die Software PVsyst kurz zu erklären.

„PVsyst ist eine Software, welche häufig verwendet wird um eine Vordimensionierung einer PV-Anlage durchzuführen. Mittels dieser Software kann ein erster Eindruck davon gewonnen werden, wie hoch die Energieproduktion im Verlaufe eines Jahres ist. Für diese Berechnung bezieht sich PVsyst auf Meteodaten der vergangenen Jahre. (...) Um die Berechnung durchzuführen, muss der Standort der Panels sowie deren Ausrichtung bezüglich der Himmelsrichtung angegeben werden. Als zusätzliche Option besteht die Möglichkeit Objekte zu platzieren, welche Schatten auf die Panels erzeugen können. Bei der Auswahl der Panels stellt PVsyst eine breite Palette mit allen gängigen PV-Panels zur Verfügung“ (Bittel, 2015).

Als Resultat dieser Berechnung erhält man für jeden Monat die theoretisch möglichen Energiewerte, welche produziert werden können. Diese werden in der Jahreskurve dargestellt, damit man einen direkten Vergleich der Ist- und der Soll-Werte erhält.

Abbildung 6.29: Simulationsdaten anzeigen

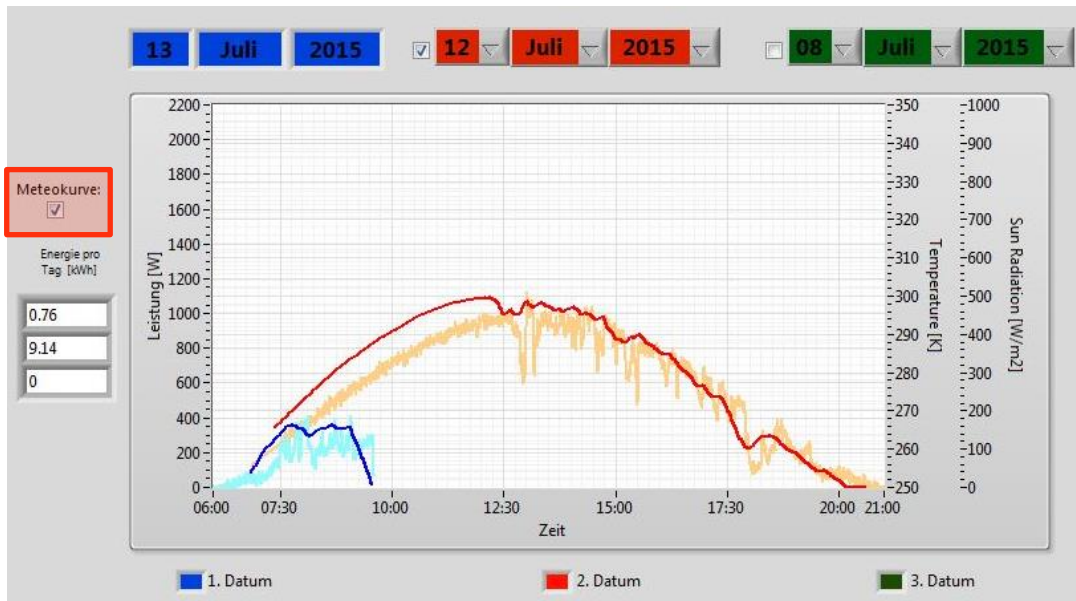


Im Graphen ist erkennbar, dass ab ca. Mitte April der theoretische Wert tiefer liegt als der aktuelle Wert. Es gibt verschiedene Typen von Solarzellen: 10 Stück sind von Inventux, 12 von CSun. Aufgrund von Testzwecken sind die Panels von CSun jedoch nicht angeschlossen. Sie werden aber nach Abschluss dieser Tests verbunden.

6.1.4.8 Meteokurve

Beim Tagesgraphen kann zusätzlich zur Kurve der generierten Energie jeweils eine Meteokurve angezeigt werden, indem man diese mit der Checkbox (rot markiert) aktiviert. Die entsprechenden Daten werden von der Meteostation gesammelt und gespeichert, welche sich auf dem Dach der HES-SO in Sion befindet.

Abbildung 6.30: Meteokurve im Tagesgraph

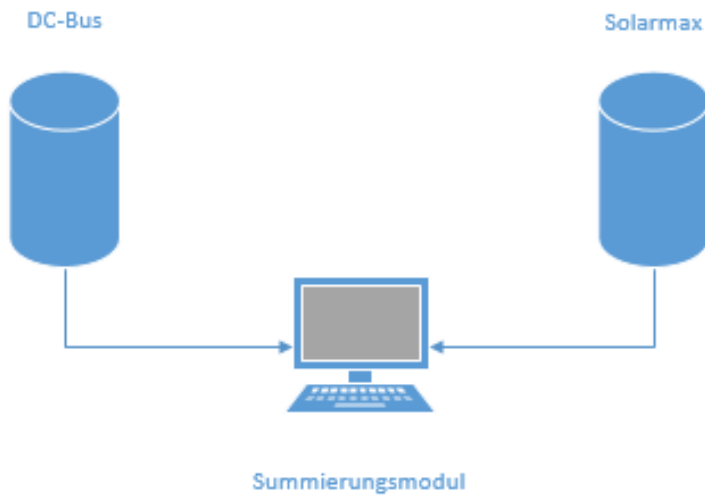


Die dunkelblaue Kurve zeigt die produzierte Energie des aktuellen Tages an (hier 13. Juli 2015), die hellblaue Kurve zeigt die Meteokurve dieses Tages und somit, was die Anlage vom Wetter her produzieren sollte. Zusätzlich wurde der 12. Juli 2015 ausgewählt, welcher die rote Kurve darstellt. Die orange Kurve ist wiederum die Meteokurve für dieses Datum.

6.2 Register Summierungsmodul

In folgendem Abschnitt geht es darum, die beiden Programme der Anlagen (DC-Bus und Solarmax) in einem sogenannten Summierungsmodul zusammenzuführen.

Abbildung 6.31: Diagramm Summierungsmodul

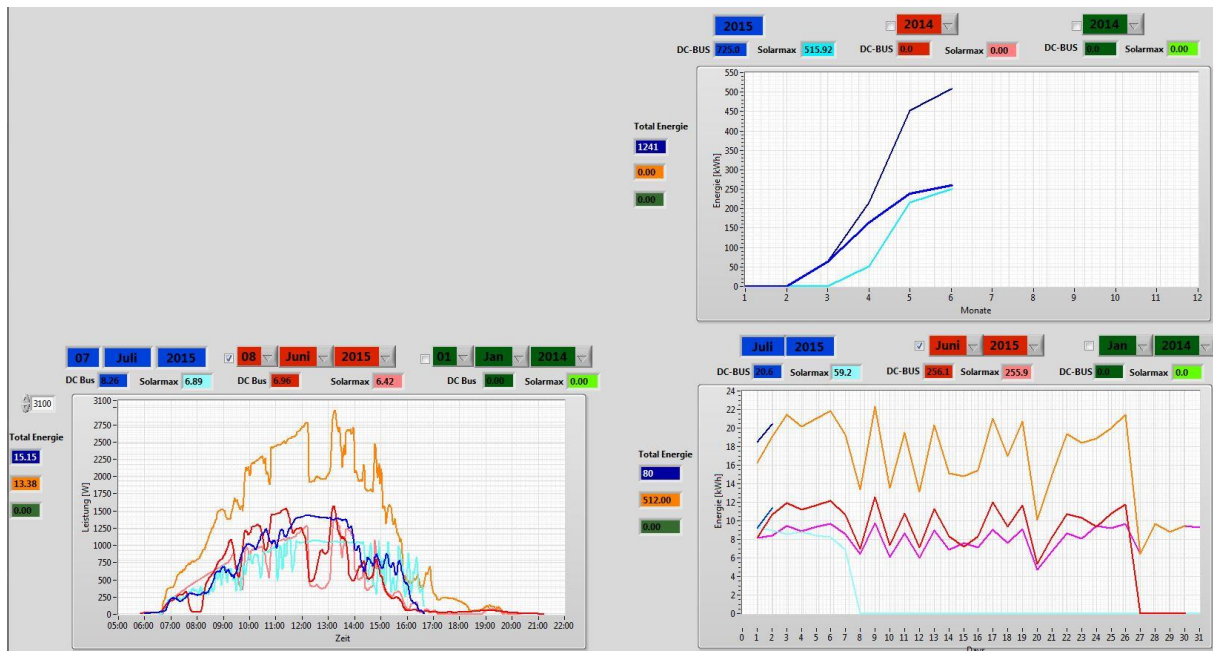


Quelle: Eigene Darstellung mithilfe von Microsoft Visio

Das Ziel ist es, in einer zusätzlichen Registerkarte die Energie beider Anlagen anzuzeigen. Diese werden wiederum jeweils in einer Tageskurve, Monatskurve wie auch in einer Jahreskurve grafisch dargestellt. Da der DC-Bus von Ivan Bittel implementiert wurde, war es der gemeinsame Auftrag, dieses erweiterte Modul zu implementieren. Seine Aufgabe bestand hauptsächlich darin, die Logik zu erstellen, damit danach die verschiedenen Graphen gemäß Abbildung 6.32 angeordnet und beide Projekte (DC-Bus und Solarmax) in einem einzigen Programm mit verschiedenen Registern zusammengefasst werden kann.

Nachfolgend wird das gesamte Register für das Summierungsmodul dargestellt. Von der Benutzeroberfläche her sieht dieses ähnlich aus wie beim Solarmax.

Abbildung 6.32: Summierungsmodul Übersicht

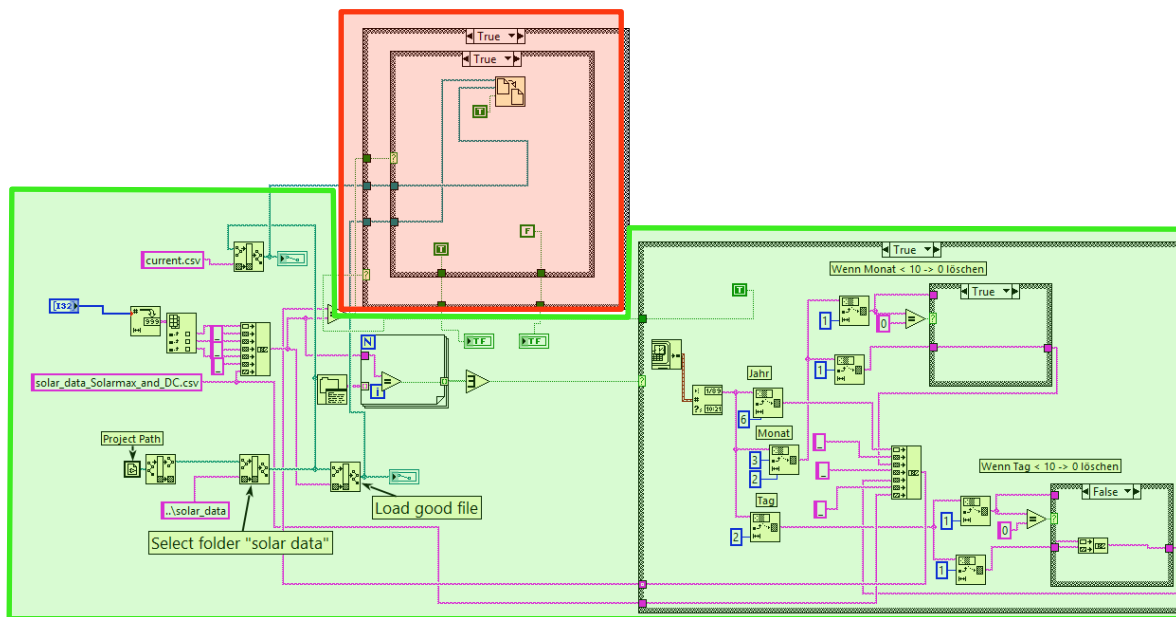


In den nächsten drei Unterkapiteln werden alle drei Graphen erläutert:

6.2.1 Summierungsmodul Tageskurve

Der schwierigste Teil des Summierungsmoduls war die Umsetzung der Summierung beider Tageskurven. Wie auch beim Tagesgraphen des Solarmax bestand das Problem darin, dass beim Ausführen des Programmes die .csv-Datei des aktuellen Tages erstellt wird und danach nicht mehr aktualisiert wird. Um dieses Problem zu lösen, musste folgende Logik eingebaut werden:

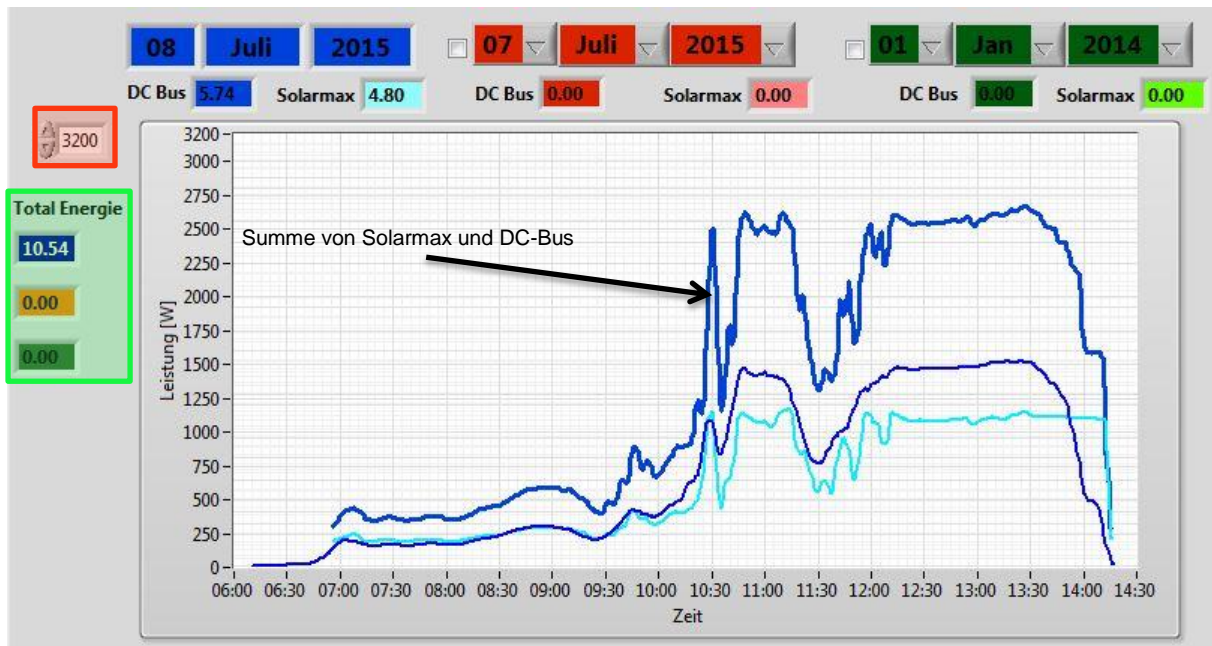
Abbildung 6.33: Summierungsmodul Logik aktueller Tag



- Grün:** Dieser Bereich ist vergleichbar mit dem in Abbildung 6.28 erklärten Ausschnitt.
- Rot:** Wenn die Datei vom aktuellen Tag ist, wird der Inhalt in eine neue .csv-Datei kopiert (current.csv) und die alte Datei gelöscht. Anschliessend wird die Datei neu erstellt, da sie nicht mehr existiert. Immer wenn die Daten komplett geladen sind, wird der Inhalt wieder kopiert. Damit wird erreicht, dass die Daten stets aktualisiert werden und somit auch der Graph immer aktuell ist. Beim Summierungsmodul musste darum auf diese komplizierte Methode (roter Bereich) zurückgegriffen werden, da das Überschreiben der Datei zuvor nicht funktionierte.

Der Tagesgraph für den aktuellen Tag sieht dann wie folgt aus:

Abbildung 6.34: Summierungsmodul Tageskurve aktueller Tag



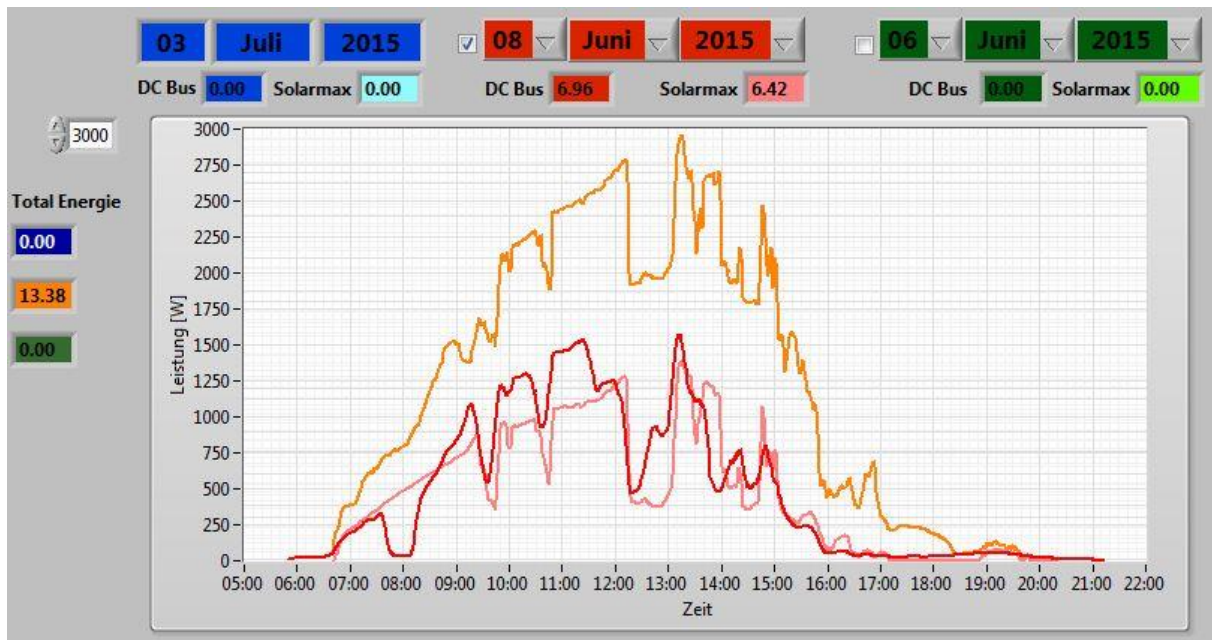
Der Aufbau des Graphen ist ähnlich wie beim Solarmax. Es können wiederum zusätzlich zum aktuellen Tag zwei frei wählbare Tage ausgewählt werden, welche aktiviert werden müssen, damit sie im Graphen erscheinen. In der Abbildung 6.34 werden nur die Werte des aktuellen Tages angezeigt, da die zusätzlichen Tage nicht aktiviert sind. Unter dem jeweiligen Datum sind immer die Anlagen DC-Bus und Solarmax mit der jeweiligen Farbe angegeben, in welcher sie im Graphen erscheinen. Die Daten werden aus beiden Datenbanken (MySQL und influxDB) ausgelesen und in den entsprechenden Farben angezeigt. In einer dritten Kurve (hier dunkelblau) werden diese zwei zusammenaddiert und im selben Graphen dargestellt.

Links neben dem Graphen (grüner Bereich) wird für jedes Datum die jeweilige Summe angegeben. Für den aktuellen Tag in der Abbildung, dem 08. Juli 2015, ist die Summe der generierten Energie bis zum Abrufzeitpunkt (um ca. 14:15 Uhr) 10.54 kWh.

Mit dem rot markierten Element kann die Y-Achse im Graphen benutzerdefiniert angepasst werden, indem man den gewünschten maximalen Wert der Achse angibt.

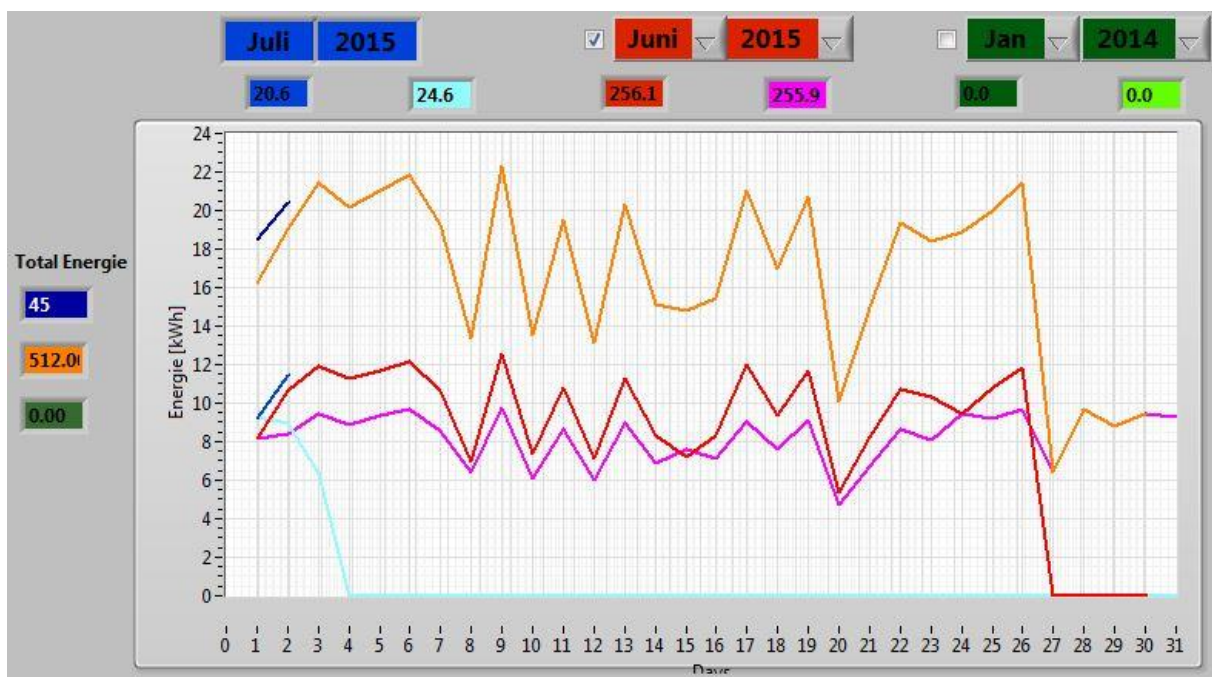
Um zur Erklärung eine gewisse Übersichtlichkeit zu gewähren, wurde in der obigen Abbildung nur der aktuelle Tag angezeigt. Nachstehend wird zusätzlich die Summierungskurve vom 08. Juni 2015 aufgezeigt.

Abbildung 6.35: Summierungsmodul Tageskurve beliebiges Datum



6.2.2 Summierungsmodul Monatskurve

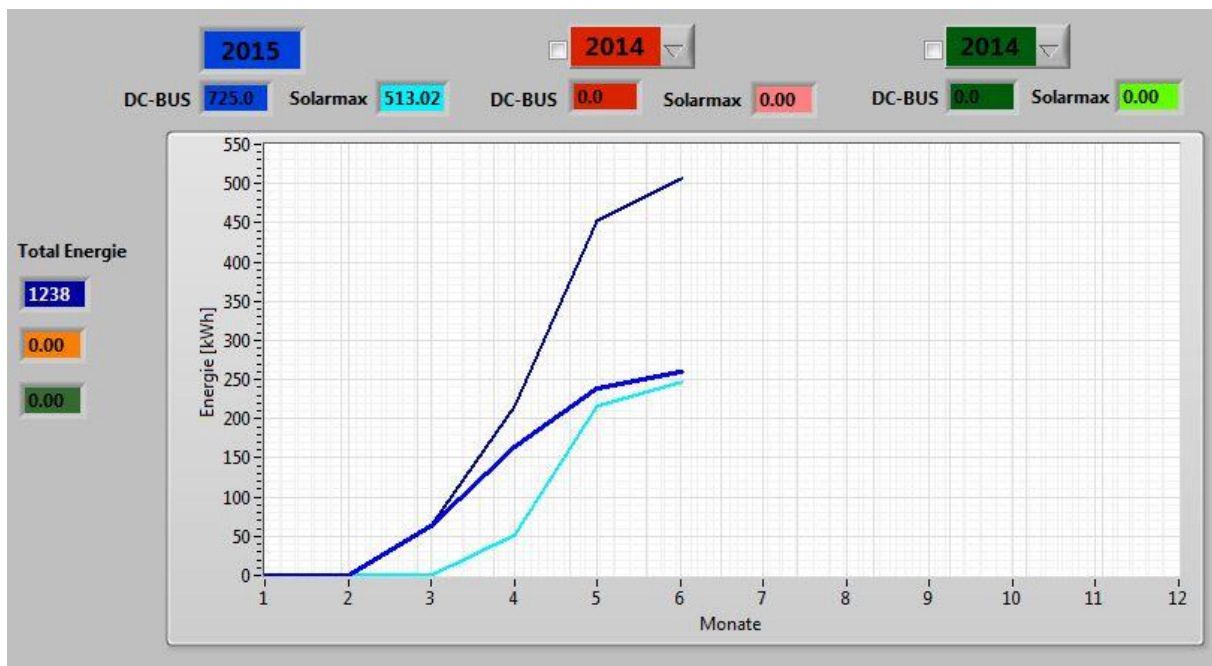
Abbildung 6.36: Summierungsmodul Monatskurve



Wie bei der Tageskurve des Summierungsmoduls werden auch bei der Monatskurve die Daten vom Solarmax (lila) und dem DC-Bus (rot) addiert und in einer dritten Kurve (orange) dargestellt. Auch hier können zwei frei wählbare Monate ausgewählt und angezeigt werden.

6.2.3 Summierungsmodul Jahreskurve

Abbildung 6.37: Summierungsmodul Jahreskurve



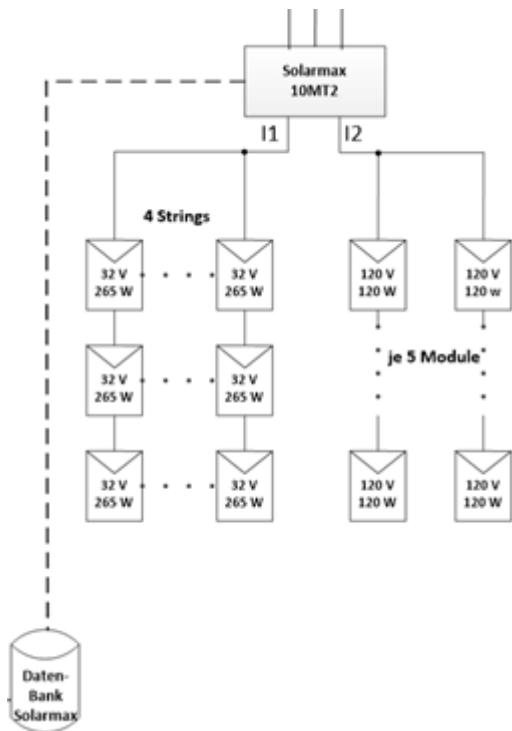
Am letzten Tag des aktuellen Monats werden die Werte dieses Monats von beiden Anlagen in der Jahreskurve ergänzt und die Summe gebildet. Zusätzlich sind wiederum zwei frei wählbare Jahre wählbar.

6.3 Register Fachmann

Für den Normalnutzer sind die Informationen über die generierte Energie pro Tag, pro Monat und pro Jahr meist ausreichend, da ihn nur interessiert, wie viel seine Anlage produziert. Bei auftretenden Fehlern sind allerdings mehr Details nötig, um den Fehler zu erkennen. Ein möglicher Fehler kann z.B. sein, wenn die Anlage nicht die Menge produziert, welche sie sollte. Daher wurde das erweiterte Register Fachmann erstellt, welches detailliertere Daten anzeigt.

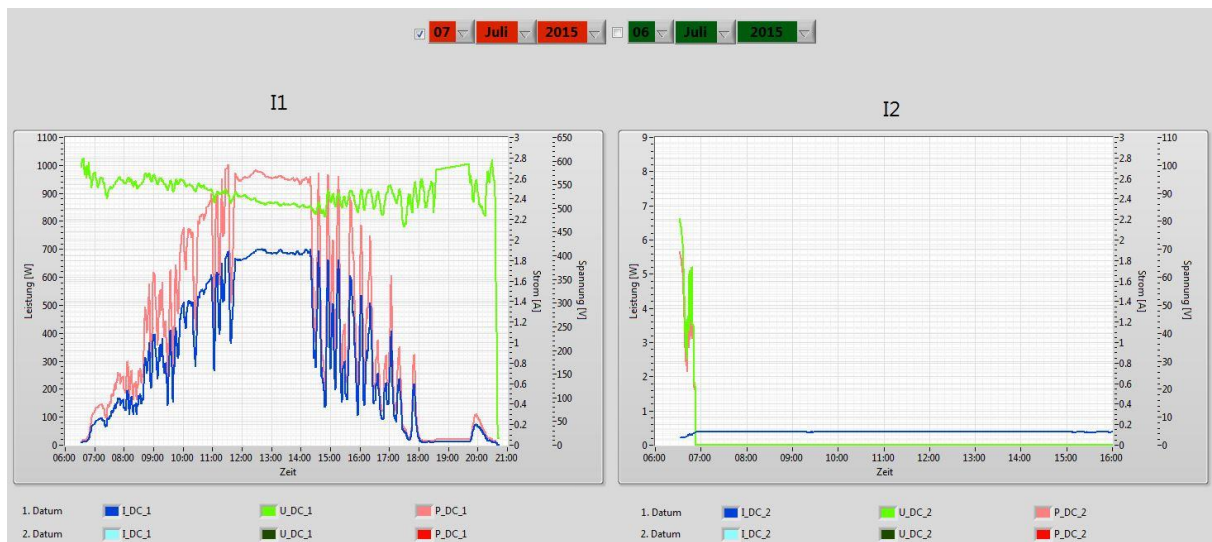
Um dies ein wenig mehr zu erläutern, wird das Schema des Solarmax noch einmal aufgezeigt:

Abbildung 6.38: Schema Solarmax



Der Solarmax hat einen Input I1 wie auch einen Input I2. Da die Panels in Serie zueinander verbunden sind und nur an einem Wechselrichter angeschlossen sind, kann man nicht jedes Panel einzeln auswählen. Die nachstehende Abbildung zeigt die Registerkarte Fachmann:

Abbildung 6.39: Übersicht Register Fachmann



In diesem Register können nur zwei frei wählbare Tage ausgewählt werden, ohne dass bereits die Werte des heutigen Datums fix ausgelesen werden. Grund dafür ist, dass pro Tag bereits drei Kurven gezeichnet werden (Strom, Spannung und Leistung) und der Graph ansonsten zu überfüllt und unübersichtlich wird. Der Graph hat mehrere Y-Achsen damit drei unterschiedliche Einheiten angezeigt werden können. In den Einstellungen ist sehr wichtig, dass man die entsprechenden Linien der richtigen Achse zuordnet, also die blauen Linien der Strom-Achse, die grünen Linien der Spannung-Achse und die roten Linien der Leistung-Achse.

In der obigen Abbildung 6.39 werden jeweils Strom I (blau), Spannung U (grün) und Leistung P (rot) des gewählten Datums angezeigt. Hier sind die nötigen Werte, die benutzt werden, um die Leistung zu berechnen, nämlich anhand der Formel Leistung $P = \text{Spannung } U * \text{Strom } I$. Wenn die produzierte Energie zu tief ist, kann anhand dieser Grafik analysiert werden, wo das Problem liegt und folglich erkannt werden, ob das Defizit durch den Strom oder die Spannung verursacht wird, oder gar andere Ursachen hat.

7. Testphase

Die Testphase verlief parallel zur Implementierungsphase. Jede Funktion die erstellt wurde, konnte direkt anhand der Live-Daten getestet werden. Es war auch möglich, von zu Hause aus zu arbeiten. Es musste lediglich das HES-SO Valais Tool gestartet werden, damit auf den HES-SO Server zugegriffen werden konnte. Das Laden der Daten dauerte jedoch mit dieser Verbindung länger anstatt mit der direkten Wireless-Verbindung in der Schule.

Die Funktionsweise des Summierungsmoduls musste am GridLab-PC getestet werden, da auf dem privaten Computer der Treiber der SQL-Datenbank für die DC-Bus-Anlage nicht installiert wurde.

8. Schlussfolgerung

Das bestehende Programm konnte an die gewünschten Anforderungen angepasst werden. Somit konnte ein erfolgreiches Programm implementiert werden, welches den Solarmax wie auch den DC-Bus impliziert. Mithilfe dieses Überwachungsprogrammes ist es auch für Privatanutzer möglich, die generierte Energie seiner PV-Anlage auf einem Display grafisch anzeigen zu lassen.

Für die Datenbankanbindung musste statt geplanten zwei Tagen mehr als zwei Wochen aufgewendet werden, da für die Solarmax-Anlage statt einer MySQL-DB eine influxDB verwendet werden musste.

Voraussetzung für das Zusammenführen der beiden Anlagen DC-Bus und Solarmax war, dass die Programme fertig implementiert waren. Darum war es wichtig, Absprachen zu halten und die Implementierungen zeitlich zu koordinieren.

Literaturverzeichnis

- Bittel, I. (07 2015). Überwachung & Optimierung einer komplexen PV Anlage. Sion.
Centralschweizerische Kraftwerke AG. (2014). *Solaranlage einfach erklärt*. Abgerufen am 18. Juli 2015 von <http://sms.ckw.ch/content/ckwsms/de/startseite/mittelstufe/solaranlage-erklaert.html>
- Dr. Eicher und Pauli AG. (2014). *Schweizerische Statistik der erneuerbaren Energien*. Abgerufen am 2015 von http://www.bfe.admin.ch/themen/00526/00541/00543/?dossier_id=00772&lang=de
- Einführung in JSON*. (kein Datum). Abgerufen am 17. Juli 2015 von <http://json.org/json-de.html>
- Häberlin, H. (2010). *Photovoltaik Strom aus Sonnenlicht für Verbundnetze und Inselanlage*. Electrosuisse Verlag.
- National Instruments Corporation. (kein Datum). *National Instruments*. Abgerufen am 12. Juli 2015 von <http://switzerland.ni.com/>
- Schweizerische Energie-Stiftung. (2014). *Schweizerische Energiestiftung*. Abgerufen am 18. Juli 2015 von <http://www.energiestiftung.ch/energiethemen/erneuerbareenergien/sonne/solarstrom/>
- Söbbing, H. (2014). *Unix Timestamp*. Abgerufen am 19. Juli 2015 von <http://www.unixtimestamp.de/>
- Wesselak, V., & Sebastian, V. (2012). *Photovoltaik - Wie Sonne zu Strom wird*. Deutschland: Springer Verlag.

Anhang I: Daten der Bachelorarbeit**HES-SO Valais**

FEE	FIG	FTO
	X	

Données du travail de bachelor
Daten der Bachelorarbeit

FO.2.2.02.28.EB
 mob/30/10/2013

Fillière / Studiengang : Informatique de gestion PT

Confidentiel / Vertraulich

Etudiant-e / Student-in NOM / NAME WYER Tél. +41 79 360 75 91	Prénom / Vorname Kevin	Année / Jahr 2015
Proposé par / vorgeschlagen von <i>Christoph Ellert & David Wannier</i> Prof. responsable du suivi : GENOUD Dominique		Langue d'exécution / Ausführungssprache Français Deutsch English

Titre / Titel: Evaluation of the photovoltaic power plant of the GridLab

Description / Beschreibung

The photovoltaic power plant of the GridLab laboratory of the School of Engineering of the HES-SO Valais-Wallis is used to test the main technical components of photovoltaic power plants.

The aim of this Bachelor's thesis is to develop and implement a management software to view and evaluate the measurement values that are characteristic of the performance of a PV plant and its components (including solar panels, inverters from different suppliers or developed in-house, battery status, weather station and the performance ratio of the entire plant).

These values are characteristic of the real performance and must be compared with the simulated values from a standard market software typically used to build PV plants. The data to evaluate is available in a centralised database or must be extracted from the components used in the PV plant.

This software needs to be developed with LabVIEW in order to be integrated into the context of the GridLab.

FEC	FIG	FTO
	X	

Objectifs / Ziele:

- Handling of the development tool LabVIEW
- Analysis of the criteria, data and measurements available to evaluate the components, definition of adequate measurements
- State of the art of simulation software, comparison of results with simulations of selected programmes
- Development of the operational evaluation software and deployment to the
- Testing of the software with the data from the database or the data extracted from components used in the PV plant
- Documentation, handbook

Commentaires du professeur / Anmerkung des Dozents

- in collaboration with Prof. Ellert of HES-SO/ISI to establish the product backlog

Signature ou visa / Unterschrift oder Visum

Resp. de la filière Informatique de gestion



Professeur/Dozent: GENOUD Dominique

Etudiant/Student: Kevin Wyer

Délais / TermineAttribution du thème / Ausgabe des Auftrags:
13.04.2015Remise du rapport / Abgabe des Schlussberichts:
27.07.2015 12h00Exposition publique / Ausstellung Bachelorarbeiten
11.09.2015

Un travail de bachelor, non confidentiel, ayant obtenu une note supérieure ou égale à cinq [Evaluation C] est publié dans son intégralité (sauf annexes pour la filière économie d'entreprise et informatique de gestion) à la médiathèque de la HES-SO Valais [RERO DOC].

Nicht vertrauliche Bachelorarbeiten, die mindestens die Note 5.0 erzielt haben (Note C), werden vollständig (ohne Anhänge für die Studiengänge Betriebsökonomie und Wirtschaftsinformatik) in der Mediathek der HES-SO Valais (RERO DOC) veröffentlicht.

Anhang II: Daily report

Woche	Tag	Datum	Stunden	Task	
1	Montag	27.04.2015	7,00	Erste Eindrücke sammeln von LabView via Homepage von National Instruments	30,50
	Dienstag	28.04.2015	1,50	Erstes Meeting mit Dominique Genoud Erstes Meeting mit Christoph Ellert und Kennenlernen von Ivan Bittel	
	Mittwoch	29.04.2015	10,00	Installieren von LabView Kennenlernen von LabView via tutorials erster Einblick in Code	
	Donnerstag	30.04.2015	6,00	Kennenlernen von LabView via tutorials	
	Samstag	02.05.2015	6,00	Kennenlernen von LabView via tutorials	
2	Montag	04.05.2015	4,50	Erste Version des Product Backlogs	26,50
	Dienstag	05.05.2015	3,50	Erste Version des Product Backlogs	
	Mittwoch	06.05.2015	9,50	Meeting mit Ivan Bittel, um Fragen über Code zu besprechen Analysieren des Codes von Ivan Bittel Kennenlernen von LabView via tutorials	
	Donnerstag	07.05.2015	6,00	Kennenlernen von LabView via tutorials Meeting mit Christoph Ellert, um Product Backlog zu besprechen	
	Freitag	08.05.2015	3,00	Meeting Dominique Genoud (Productbacklog ok?) Product Backlog fertig stellen	
3	Montag	11.05.2015	9,50	Bestehendes Projekt ausdrucken, zusammensetzen und analysieren Infos über Photovoltaik-Anlagen sammeln	29,50
	Dienstag	12.05.2015	8,50	Bestehendes Projekt bearbeiten und lernen zu verstehen Meeting mit Christoph Ellert 13:30 Code auf Computer in GridLab laufen lassen Kennenlernen von LabView mit Buch	
	Mittwoch	13.05.2015	8,50	Kennenlernen von LabView mit Buch Bestehendes Programm analysieren und erste Anpassungen	
	Freitag	15.05.2015	3,00	Grundgerüst der Dokumentation erstellen, und schreiben von der Einleitung	

4	Montag	18.05.2015	9,25	Datenblatt erstellen, welche Daten aus Solarmax geladen werden Mit Herrn Ellert besprechen, wie die Wh von SolarMax berechnet wird Meeting Michael Clausen wegen influxDB Verbindung	32,75
	Dienstag	19.05.2015	9,50	Meeting mit Christoph Ellert 13:30 Nach Lösung suchen, wie Datenbankverbindung mit influxDB hergestellt wird	
	Mittwoch	20.05.2015	10,50	Funktionen verstehen lernen, welche Daten aus Datenbank ausgelesen werden Meeting mit Michael Clausen (Zugriff auf Grafana) Kennenlernen von Grafana und influxDB Installieren von i3 JSON Toolkit Versuch von Verbindungsaufbau mit influxDB in LabView	
	Donnerstag	21.05.2015	3,50	Kennenlernen von Grafana und influxDB Meeting mit Dominique Genoud, um aktuellen Stand zu besprechen Versuch von Verbindungsaufbau mit influxDB in LabView	
5	Montag	25.05.2015	10,00	Verbindungsaufbau mit influxDB in LabView mit HTTP Client Parsen von JSON string nach LabView String (Probleme mit Klammern[])	35,50
	Donnerstag	28.05.2015	10,00	Parsen von JSON string nach LabView String (Probleme mit Klammern[]) Problembeschreibung in der Dokumentation	
	Freitag	29.05.2015	10,50	Parsen von JSON string nach LabView String war nicht möglich -> mit normalen String Funktionen (Substring, Index String, Concatenate...) Daten aus der Datenbank in LabView schön strukturieren und in .csv Datei speichern	
	Samstag	30.05.2015	5,00	Daten aus der Datenbank in LabView schön strukturieren und in .csv Datei speichern	
6	Montag	01.06.2015	4,00	Funktion verstehen lernen, um Daten aus csv zu laden und grafisch darzustellen	32,00
	Dienstag	02.06.2015	5,50	Meeting Ellert 13:30 Sub-VI, die ich erstellt habe um Daten aus Datenbank in csv zu speichern, in bestehendes Programm einbinden	
	Mittwoch	03.06.2015	9,00	BA meeting 08:30 Daten von beliebigem Datum in csv speichern	
	Freitag	05.06.2015	8,50	Daten von beliebigem Datum in csv speichern mit richtigem Zeitstempel (Sekunden seit Mitternacht) Anzeigen der Daten in Graph Anzeigen von Leistung total	
	Samstag	06.06.2015	5,00	Daten von aktuellem Tag im Graph laufend aktualisieren	

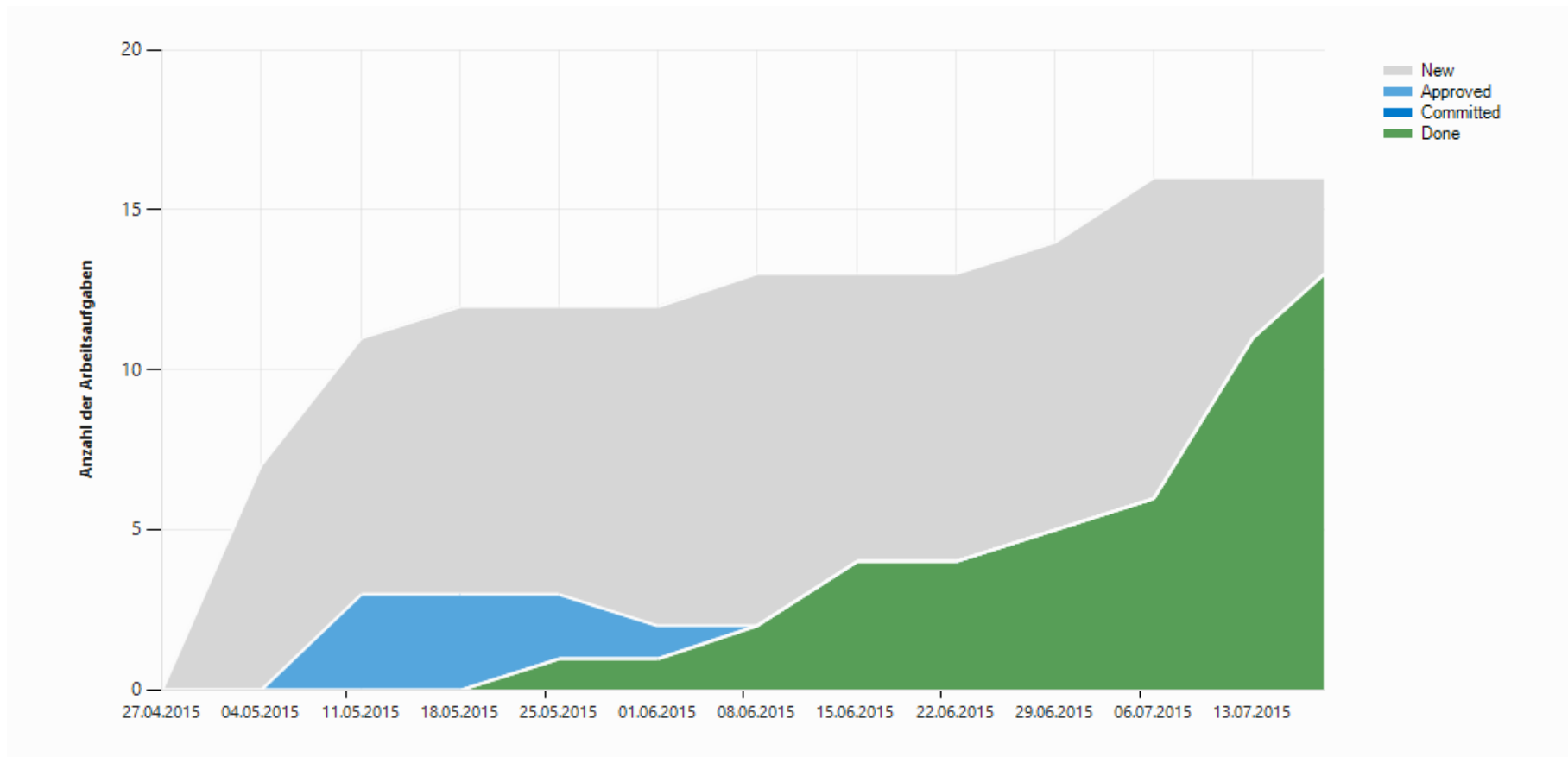
7	Montag	08.06.2015	7,00	Daten von aktuellem Tag im Graph laufend aktualisieren (konnte nicht getestet werden: Server down)	39,25
	Dienstag	09.06.2015	9,00	Daten von aktuellem Tag im Graph laufend aktualisieren Vorbereitung Meeting Meeting Genoud 14:00	
	Mittwoch	10.06.2015	8,25	Daten von aktuellem Tag im Graph laufend aktualisieren Generierte kWh pro Tag addieren und anzeigen Daten von aktuellem Monat im Graph laufend aktualisieren Daten von gewähltem Monat im Graph anzeigen	
	Donnerstag	11.06.2015	9,00	Anpassungen, damit Datum um Mitternacht angepasst wird Jahreskurve wird falsch dargestellt (Problem analysieren) 16:00 Meeting Ellert	
	Freitag	11.06.2015	6,00	Gesamten Ablauf der Funktion für Jahreskurve analysieren	
8	Montag	29.06.2015	9,00	Funktion der Jahreskurve überarbeiten	53,00
	Dienstag	30.06.2015	10,00	Funktion der Jahreskurve überarbeiten Am letzten Tag des Monats um 23.30 Summe der Monatsdaten in Jahresdatei einfügen Meeting Ellert 13:30 Schema von Solarmax in LabView zeichnen Flussdiagramm zeichnen	
	Mittwoch	01.07.2015	9,00	Meteo Datenbankverbindung herstellen Meteo Kurve von ausgewähltem Datum anzeigen	
	Donnerstag	02.07.2015	9,50	Summierungsmodul (Tageskurve) implementieren mit Ivan Bittel Meeting Genoud 14:00 Product Backlog restliche Zeit berechnen	
	Freitag	03.07.2015	9,50	Summierungsmodul (Tageskurve) implementieren Summierungsmodul (Monatskurve) implementieren	
	Samstag	04.07.2015	6,00	Fehler beheben in der Monatskurve (Daten wurden nicht mehr aktualisiert)	

9	Montag	06.07.2015	5,00	Summierungsmodul (Jahreskurve) implementieren	51,00
	Dienstag	07.07.2015	10,00	Meeting mit Genoud und Ellert 08:30 Tab 'Fachmann' implementieren	
	Mittwoch	08.07.2015	9,00	Tab 'Fachmann' implementieren Tagesgraph Summierungsmodul aktueller Tag Probleme beheben	
	Donnerstag	09.07.2015	9,00	Solarmax und Summierungsmodul in ein Projekt zusammenfügen	
	Freitag	10.07.2015	9,00	Solarmax und DC-Bus in ein Projekt zusammenfügen Tagesgraph Summierungsmodul aktueller Tag Probleme beheben	
	Samstag	11.07.2015	5,00	Solarmax und DC-Bus in ein Projekt zusammenfügen	
	Sonntag	12.07.2015	4,00	Solarmax und DC-Bus in ein Projekt zusammenfügen	
10	Montag	13.07.2015	4,50	Dokumentation schreiben	42,50
	Dienstag	14.07.2015	8,00	Dokumentation schreiben	
	Mittwoch	15.07.2015	8,00	Dokumentation schreiben	
	Donnerstag	16.07.2015	4,00	Dokumentation schreiben	
	Freitag	17.07.2015	7,00	Dokumentation schreiben	
	Samstag	18.07.2015	5,00	Dokumentation schreiben	
	Sonntag	19.07.2015	6,00	Dokumentation schreiben	
11	Montag	20.07.2015	4,00	Dokumentation schreiben	28,50
	Dienstag	21.07.2015	7,00	Dokumentation schreiben	
	Mittwoch	22.07.2015	4,00	Dokumentation schreiben	
	Donnerstag	23.07.2015	7,00	Dokumentation schreiben, überarbeiten und drucken	
	Freitag	24.07.2015	6,50	binden, auf CD speichern und Abgabe	
12	Montag	27.07.2015	-		-
Total Stunden			401,00		

Anhang III: product backlog

■ to get in LabView	Done	13	2015-GridLab\Release 1\Sprint 1
■ connection to database vlength	Done	8	2015-GridLab\Release 1\Sprint 2
■ Define Interface of sumation module	Done	2	2015-GridLab\Release 1\Sprint 3
■ Implement Solarmax part 1	Done	5	2015-GridLab\Release 1\Sprint 2
■ Implement Solarmax part 2	Done	13	2015-GridLab\Release 1\Sprint 3
■ calculate measurement values from past years	Done	5	2015-GridLab\Release 1\Sprint 3
■ Implement sumation module part1	Done	8	2015-GridLab\Release 1\Sprint 3
■ implement sumation module part2	Done	8	2015-GridLab\Release 1\Sprint 4
■ Implement tab 'Fachmann'	Done	5	2015-GridLab\Release 1\Sprint 4
■ show meteodata	Done	8	2015-GridLab\Release 1\Sprint 4
■ Simulation of PV plant	Done	2	2015-GridLab\Release 1\Sprint 4
■ write facility- and software document part 1	Done	8	2015-GridLab\Release 1\Sprint 4
■ write facility- and software document part 2	Done	13	2015-GridLab\Release 1\Sprint 5

Kummulativer Ablauf



Wyer Kevin

Sprint 1

Title	State	Assigned To	Remaining...	Effort
to get in LabView	Done	kwyer		13
install software	Done	kwyer		
watch tutorials	Done	kwyer		
analyse existing code	Done	kwyer		

Sprint 1 „to get in LabView“

Iteration 2015-GridLab\Release 1\Sprint 1

STATUS	DETAILS
Assigned To: kwyer	Effort: 13
State: Done	Area: 2015-GridLab
Reason: Work finished	Backlog Priority: 250000

DESCRIPTION STORYBOARDS TEST CASES TASKS (3)

ACCEPTANCE CRITERIA HISTORY LINKS (3) ATTACHMENTS

install LabView and learn how to work with it

Know how to work with LabView

Sprint 2

Title	State	Assigned To	Remaining...	Effort
connection to database vlangmon	Done	kwyer		8
Implement Solarmax part 1	Done	kwyer	5	5
Get Data from DB	To Do	kwyer	5	

Sprint 2 „connection to database vlengthmon“

Iteration 2015-GridLab\Release 1\Sprint 2

STATUS		DETAILS	
Assigned To	kwyer	Effort	8
State	Done	Area	2015-GridLab
Reason	Work finished	Backlog Priority	375000

DESCRIPTION	ACCEPTANCE CRITERIA
<p>connection to database vlengthmon</p> <p>receive data from influxDB: - work with a http client with the GET through the url, to get the data via SQL query as a JSON in a txt file - LabView: install i3 JSON Toolkit to handle with JSON</p>	<p>database connection is working the right way</p>

Sprint 2 „implement Solarmax part 1“

Iteration 2015-GridLab\Release 1\Sprint 2

STATUS		DETAILS	
Assigned To	kwyer	Effort	5
State	Done	Area	2015-GridLab
Reason	Work finished	Backlog Priority	562500

DESCRIPTION	ACCEPTANCE CRITERIA
<p>get data from DB influxDB</p>	<p>load the data successfully into the csv file</p>

Wyer Kevin

Sprint 3

Title	State	Assigned To	Remaining...	Effort
Define Interface of sumation module	Done	kwyer		2
Implement Solarmax part 2	Done	kwyer		13
calculate mesurement values from past years	Done	kwyer		5
Implement sumation module part1	Done	kwyer		8

Sprint 3 „Define interface of sumation module“

Iteration: 2015-GridLab\Release 1\Sprint 3

STATUS	DETAILS
Assigned To: kwyer	Effort: 2
State: Done	Area: 2015-GridLab
Reason: Work finished	Backlog Priority: 500000

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

Define Interface with SI-student, in order to integrate both data sets into a summation module

- both programs run in background for a while, then include into sumation module
- when/how to merge the software
- for daily data: problem with sumation (in which second? solarmax loads every 5 second)

sum of both plants
1 screen: schema of Ivan
1 screen: schema of Kevin
1 screen: sum of both schemas

Wyer Kevin

Sprint 3 „implement Solarmax part 2“

Iteration 2015-GridLab\Release 1\Sprint 3

STATUS	DETAILS
Assigned To: <input type="text" value="kwyer"/>	Effort: <input type="text" value="13"/>
State: <input type="text" value="Done"/>	Area: <input type="text" value="2015-GridLab"/>
Reason: <input type="text" value="Work finished"/>	Backlog Priority: <input type="text" value="625000"/>

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

DESCRIPTION

Was gibts? Die Daten des Solarmax.
Was will ich anzeigen? Alle Daten

Testen läuft permanent bis zum Ende der Arbeit

Schleifengrösse optimieren (kein leerer Platz)

ACCEPTANCE CRITERIA

the right data in the statistics

Sprint 3 „calculate measurement values from past years“

Iteration 2015-GridLab\Release 1\Sprint 3

STATUS	DETAILS
Assigned To: <input type="text" value="kwyer"/>	Effort: <input type="text" value="5"/>
State: <input type="text" value="Done"/>	Area: <input type="text" value="2015-GridLab"/>
Reason: <input type="text" value="Work finished"/>	Backlog Priority: <input type="text" value="687500"/>

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

DESCRIPTION

calculate measurement values from past years and actual year the first time, so that the loading time decrease for the next time

ACCEPTANCE CRITERIA

at midnight the values are summed, so that the statistics load faster

Wyer Kevin

Sprint 3 „implement sumation module part 1“

Iteration 2015-GridLab\Release 1\Sprint 3

STATUS		DETAILS	
Assigned To	kwyer	Effort	8
State	Done	Area	2015-GridLab
Reason	Work finished	Backlog Priority	750000

DESCRIPTION STORYBOARDS TEST CASES TASKS

ACCEPTANCE CRITERIA HISTORY LINKS ATTACHMENTS

Implementation of summation module and the module for the commercial DC/AC converter
-daily graph

the right sum of the both plants is shown

Sprint 4

Title	State	Assigned To	Remaining...	Effort
implement sumation module part2	Done	kwyer		8
Implement tab 'Fachmann'	Done	kwyer		5
show meteodata	Done	kwyer		8
Simulation of PV plant	Done	kwyer		2
write facility- and softwaredocument part 1	New	kwyer		8

Sprint 4 „implement sumation module part 2“

Iteration 2015-GridLab\Release 1\Sprint 4

STATUS	DETAILS
Assigned To: kwyer	Effort: 8
State: Done	Area: 2015-GridLab
Reason: Work finished	Backlog Priority: 785156

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

B / U

Implementation of summation module and the module for the commercial DC/AC converter

- monthly graph
- yearly graph

concatenate both programs in one project

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

B / U

the right sum of the both plants is shown

Sprint 4 „implement tab Fachmann“

Iteration 2015-GridLab\Release 1\Sprint 4

STATUS	DETAILS
Assigned To: kwyer	Effort: 5
State: Done	Area: 2015-GridLab
Reason: Work finished	Backlog Priority: 820312

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

B / U

Wie will ichs anzeigen? separiert nach Elektro-Fachmann (keine Ansicht für Solarexperte, weil anders als bei DC-Bus System keine Einzelpanels sichtbar sind. In der Fachmann-Ansicht keine Jahresdaten nötig)

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

B / U


the right data in the statistics

Sprint 4 „show meteodata“

Iteration 2015-GridLab\Release 1\Sprint 4

STATUS	DETAILS
Assigned To: kwyer	Effort: 8
State: Done	Area: 2015-GridLab
Reason: Work finished	Backlog Priority: 890625

DESCRIPTION STORYBOARDS TEST CASES TASKS


B / U 

load meteodata

Von Ivan.

Berechnung, wieviel Anlage bringen sollte, und anzeigen

ACCEPTANCE CRITERIA HISTORY LINKS ATTACHMENTS

B / U 


verify that the correct meteodata are loaded

Sprint 4 „simulation of PV plant“

Iteration 2015-GridLab\Release 1\Sprint 4


STATUS	DETAILS
Assigned To: kwyer	Effort: 2
State: Done	Area: 2015-GridLab
Reason: Work finished	Backlog Priority: 941406

DESCRIPTION STORYBOARDS TEST CASES TASKS

B / U 

- learn how to work with the software
- understand the function of power plant (MPPT maximum power point tracker)
- estimate simulation

ACCEPTANCE CRITERIA HISTORY LINKS ATTACHMENTS

B / U 

know how to work with the simulation software

Sprint 4 „write facility- and softwaredocument part 1“

Iteration

STATUS	DETAILS
Assigned To <input type="text" value="kwyer"/>	Effort <input type="text" value="8"/>
State <input type="text" value="Done"/>	Area <input type="text" value="2015-GridLab"/>
Reason <input type="text" value="Work finished"/>	Backlog Priority <input type="text" value="976562"/>

DESCRIPTION STORYBOARDS TEST CASES TASKS

B / U

- Schema
- Softwarebeschreibung

ACCEPTANCE CRITERIA HISTORY LINKS ATTACHMENTS

B / U

report should include all the recommended parts

Sprint 5 „write facility- and softwaredocument part 2“

Iteration

STATUS	DETAILS
Assigned To <input type="text"/>	Effort <input type="text" value="13"/>
State <input type="text" value="Done"/>	Area <input type="text" value="2015-GridLab"/>
Reason <input type="text" value="Work finished"/>	Backlog Priority <input type="text" value="988281"/>

DESCRIPTION STORYBOARDS TEST CASES TASKS

B / U

- Schema
- Softwarebeschreibung

ACCEPTANCE CRITERIA HISTORY LINKS ATTACHMENTS

B / U

report should include all the recommended parts

Selbständigkeitserklärung

Ich bestätige hiermit, dass ich die vorliegende Bachelorarbeit alleine und nur mit den angegebenen Hilfsmitteln realisiert habe und ausschliesslich die erwähnten Quellen benutzt habe. Ohne Einverständnis des Studiengangleiters und des für die Bachelorarbeit verantwortlichen Dozierenden sowie des Forschungspartners, mit dem ich zusammengearbeitet habe, werde ich diesen Bericht an niemanden verteilen, ausser an die Personen, die mir die wichtigsten Informationen für die Verfassung dieses Berichts geliefert haben und die ich nachstehend aufzähle:

- Herr Dominique Genoud, Hochschule für Wirtschaft & Tourismus Hes-so Sierre
- Herr Christoph Ellert, Hochschule für Ingenieurwissenschaften Hes-so Sion

Visp, 24. Juli 2015