

Smart Plug

Die intelligente Steckdose

1. Inhaltsverzeichnis

2.	Angabe	2
3.	Hardware	2
3.1.	Blockschaltbild	2
3.2.	Messplatine	3
3.3.	MCP39F501	3
3.4.	Schaltplan	6
3.5.	Kommunikation	7
3.5.1.	MODE/DIR	8
3.5.2.	RS485	8
3.6.	Versorgungsplatine	9
3.7.	Programmierung des MCP39F501	11
3.7.1.	Protokoll	13
3.8.	Bauteile	14
4.	Software	15
4.1.	Betriebssystem und installierte Programme/Bibliotheken	15
4.1.1.	Installierte Programme und Bibliotheken	15
4.1.2.	Verwendete, jedoch nicht installierte Bibliotheken	15
4.2.	Funktionsprinzip	15
4.2.1.	Blockschaltbild	16
4.3.	SmartPlug Service	16
4.4.	Datenbank	17
4.5.	Webinterface	18

2. Angabe

Die intelligente Steckdose dient dazu, Strom zu sparen und somit den Stromverbrauch zu senken. Ziel ist es, den Benutzer die einzelnen Stromfresser bzw. den Stromverbrauch der angeschlossenen Geräte zu messen und anzuzeigen.

Die graphische Auswertung soll mithilfe eines Raspberry Pi durchgeführt werden.

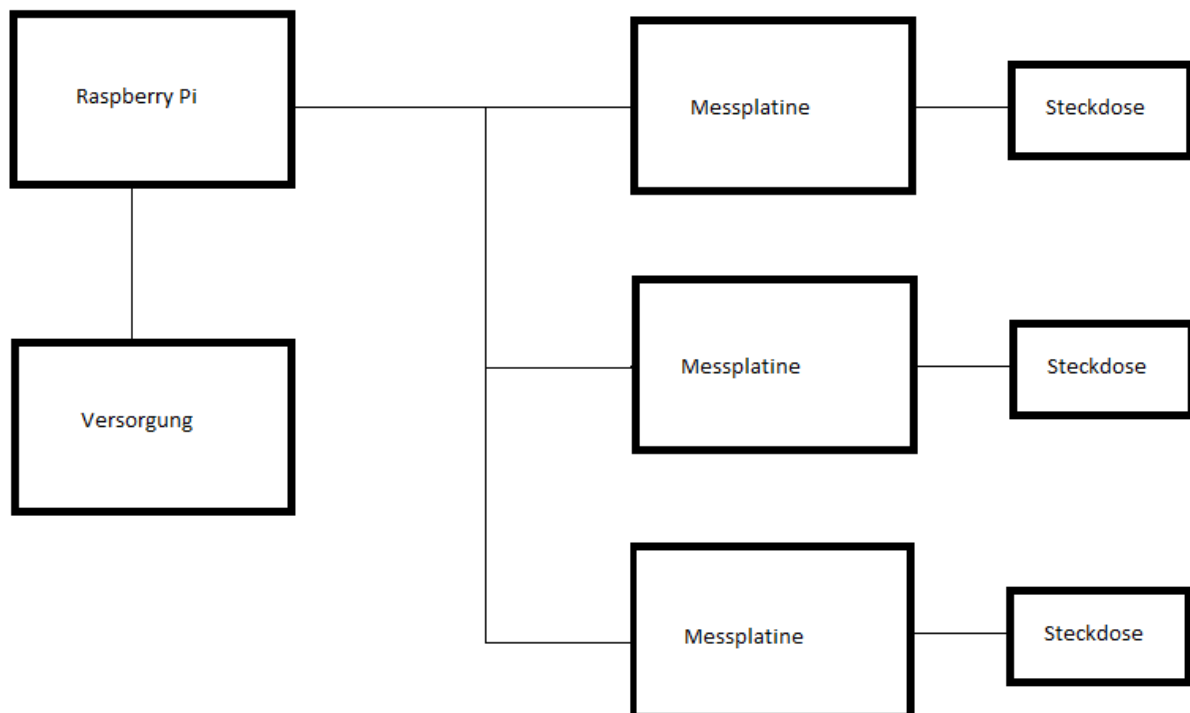
Des Weiteren sollen die Messwerte in einem Diagramm auf einem Webserver abrufbar sein.

Um den Standby-Verbrauch zu senken müssen die Steckdosen abschaltbar sein.

Die Abschaltung der einzelnen Steckdosen muss individuell Ansteuerbar sein und über den Webserver Programmierbar sein.

3. Hardware

3.1. Blockschaltbild

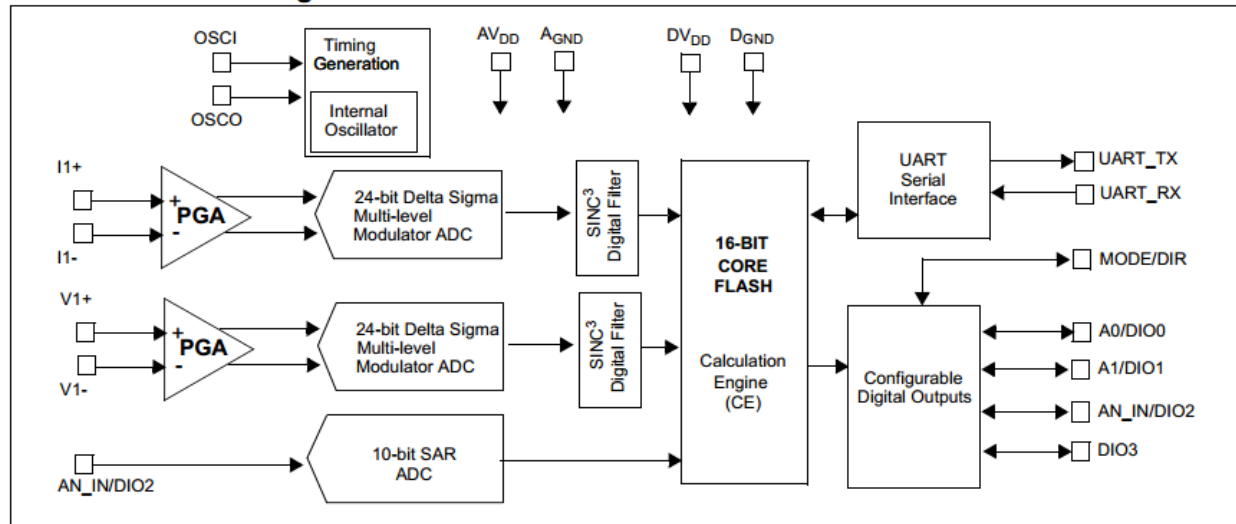


3.2. Messplatine

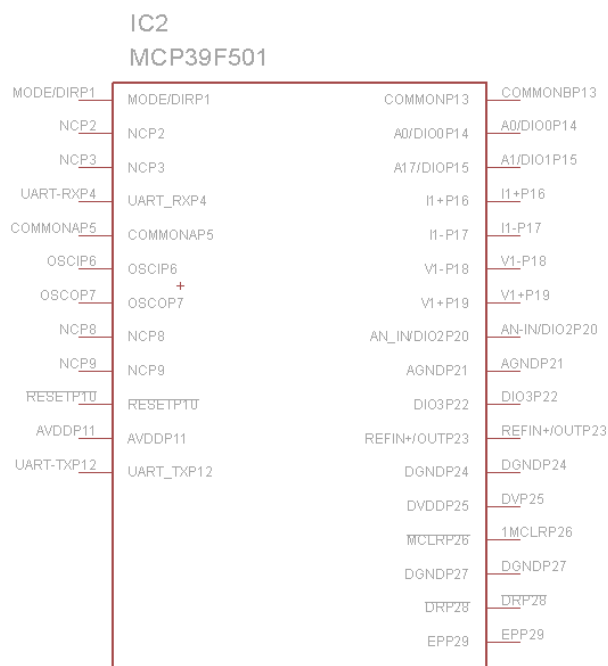
Gewählter IC zur Messung von Strom und Spannung: MCP39F501.

3.3. MCP39F501

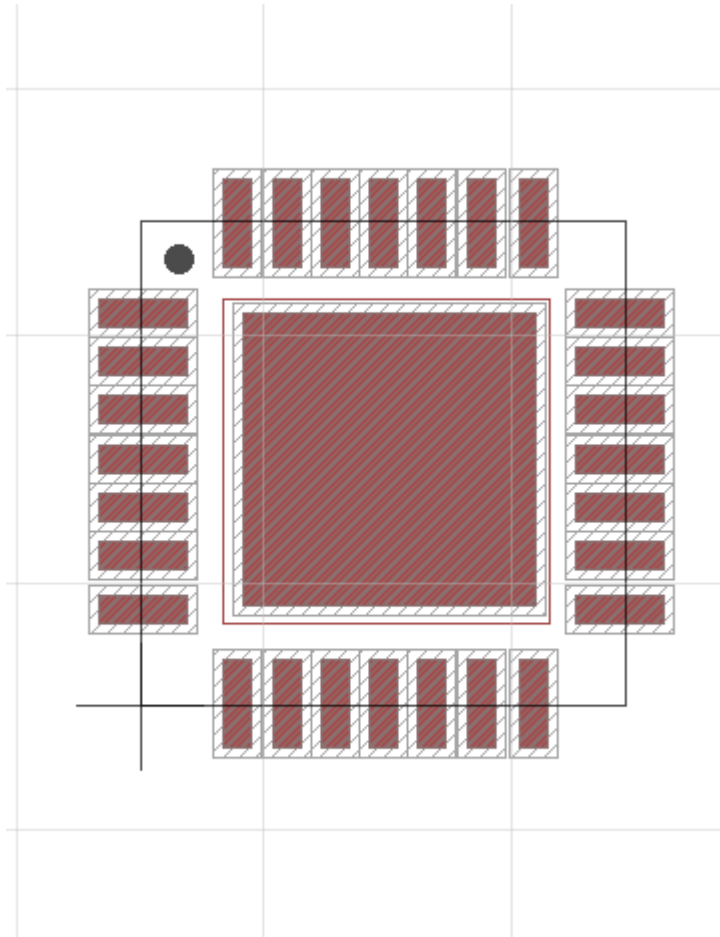
Functional Block Diagram



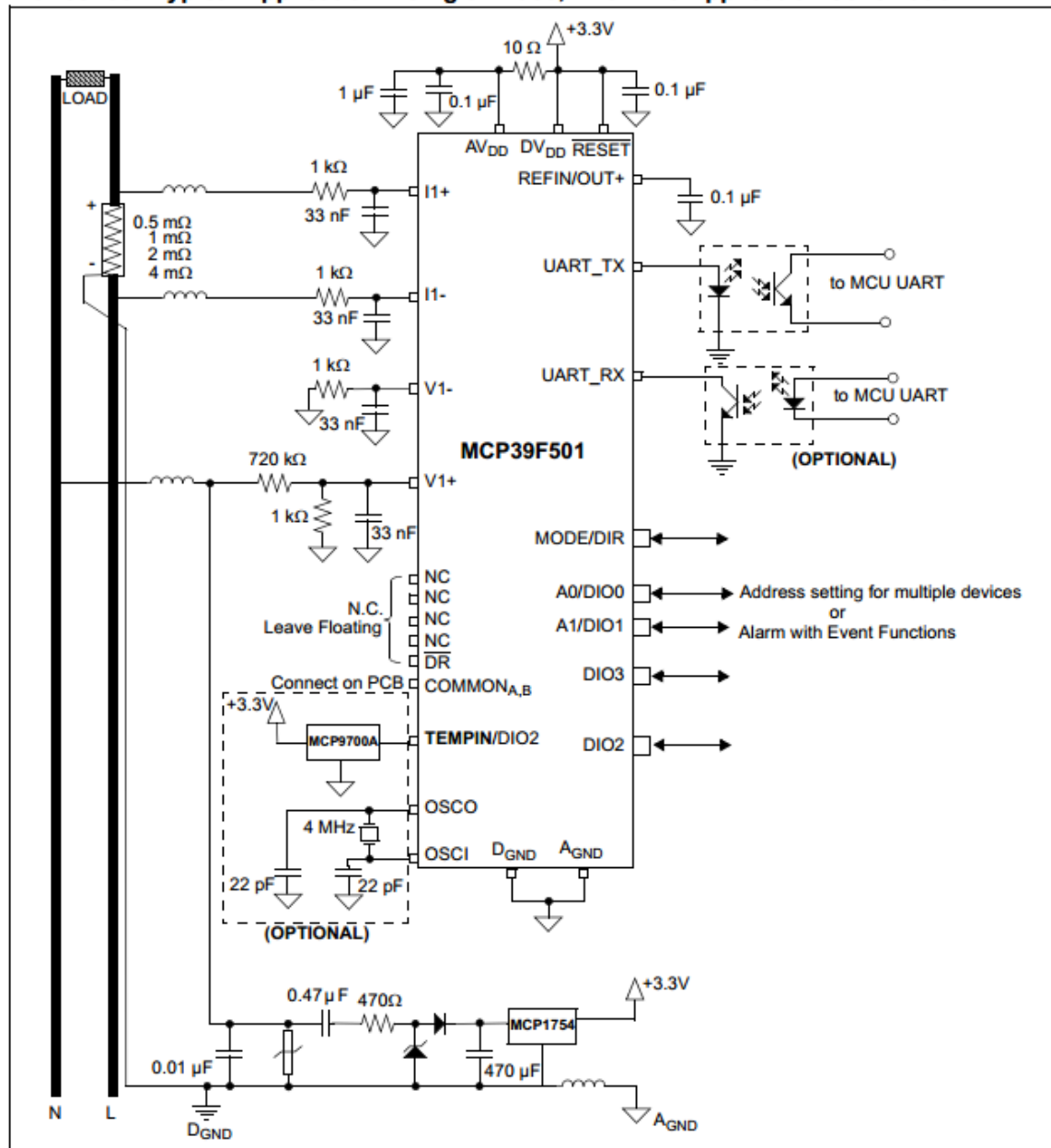
Da der MCP39F501 nicht in der Eagle-Ibr nicht vorhanden war, wurde dieser Baustein erstellt.



MCP39F501 wurde mithilfe des Datenblattes von Microchip erstellt.

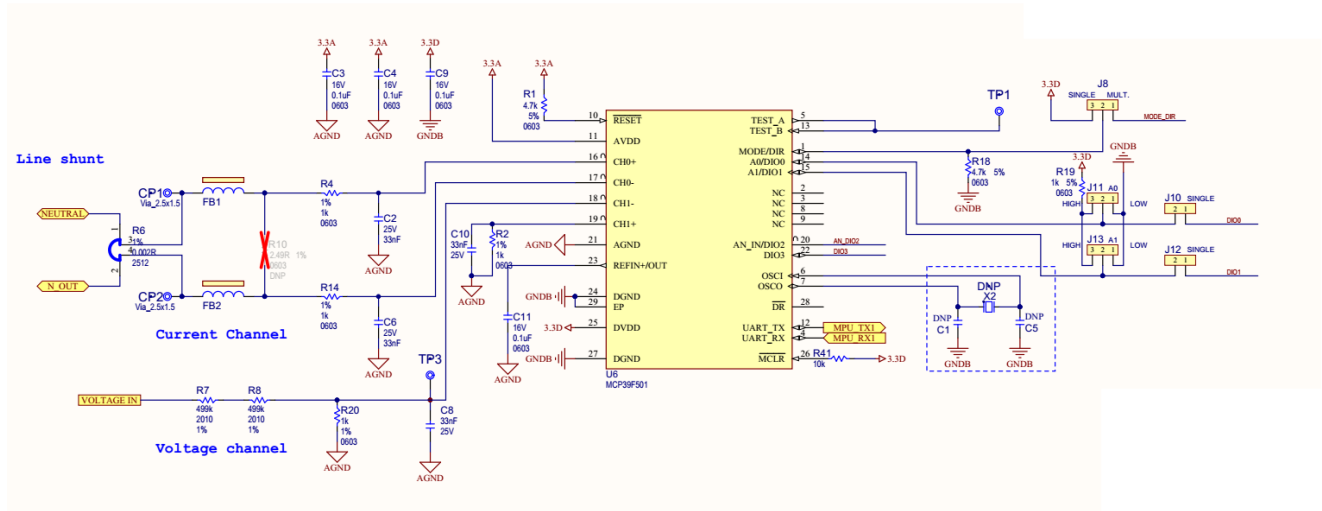


MCP39F501 Typical Application – Single Phase, Two-Wire Application Schematic

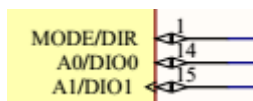


3.4. Schaltplan

Als Vorlage für den Schaltplan dient das Development Board von Microchip.



Jedoch wurde kein externer Oszillator verwendet. Des Weiteren sind die Pins 12 (UART_TX) und Pin 4 (UART_RX) auf einen MAX489CSD geführt. Der MAX489CSD wandelt das UART-Signal für einen RS485-Bus um. Somit ist es möglich, dass der RPI vier Steckdosen messen bzw. auswerten kann. Die Zahl vier ergibt sich daraus, dass der RPI zwei Pins zur Adressierung zur Verfügung stellt (Pin 14,15).

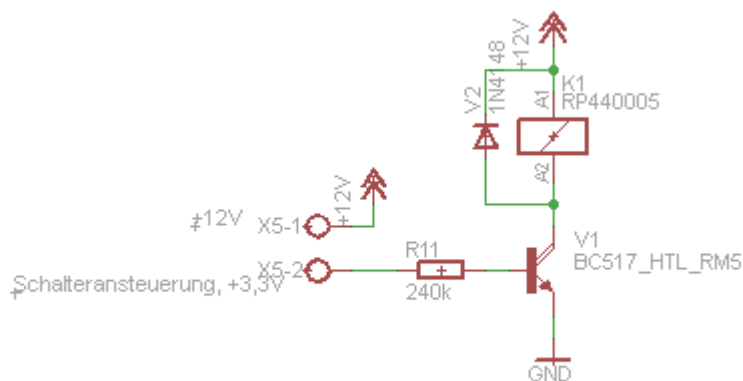


Für den Prototypenaufbau wurde der MODE/DIR-Pin, wie im Schaltplan auf einen Jumper geführt, um ihn messen zu können. Die Bauteilwerte wurden 1:1 vom Development Board übernommen.

Um die Spannung an der Steckdose ein bzw. auszuschalten wird ein Monostabiles Relais verwendet. Ein Monostabiles Relais wurde gewählt, da bei Spannungslosem Zustand das Relais offen ist. -> Steckdose Spannungsfrei bzw. Default Wert.

Die Ausgangsspannung eines GPIO beträgt 3,3V (HIGH-Zustand). Jedoch benötigt das Relais 12V für die Ansteuerung. -> Transistor als Schalter -> gewählt: BC517

Die Schaltpläne wurden in „Eagle V6.5.0“ gezeichnet und geroutet.



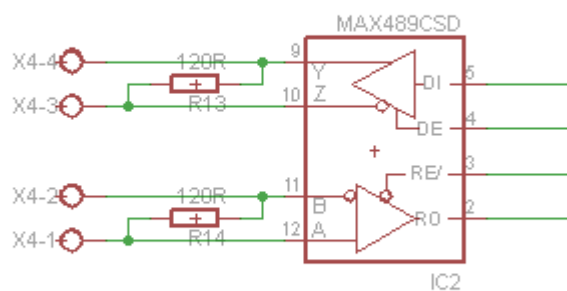
Die Diode dient als Freilaufdiode. -> Induktive Last

3.5. Kommunikation

RS485-Bussystem wurde gewählt, da dieser mehrerer Geräte unterstützt.

Für die Kommunikation werden die Pins MODE/DIR; UART_TX und UART_RX benötigt.

RS485, 4Wire



Anschlussbelegung:

MAX489CSD	MCP39F501
Pin 5	UART_TX
Pin 4	MODE/DIR
Pin 3	MODE/DIR
Pin 2	UART_RX

3.5.1. MODE/DIR

3.1 Single/Multiple Device Mode and Direction Pin (MODE/DIR)

When using multiple devices on a single bus, this pin should be tied to the DIR pin of the transceiver for direction control. This will cause the A0/DIO0 and A1/DIO1 pins to act as address pins A0,A1. If additional devices are required, the Device Address register can be programmed to allow for more than four devices. For this operation, a 4.7 k Ω pull-down resistor should be connected to this pin.

If only a single device is being used, the MODE pin should be driven high at power-on reset (POR), making the A0/DIO0 and A1/DIO1 pins additional configurable digital I/O (DIO).

4.3 Multiple Devices

In order to support multiple devices on a single bus, a device must be selected before any communication occurs between the host MCU and the individual MCP39F501 devices using the `Select Device` command. Once selected, the device will assert the DIR pin until it is deselected.

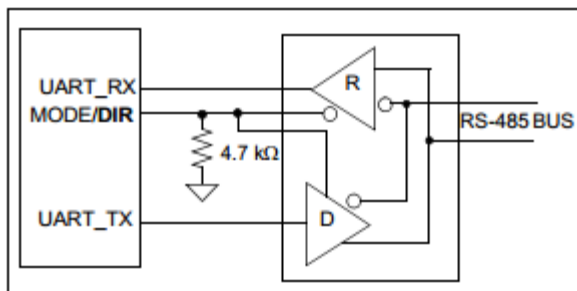


FIGURE 4-2: DIR Pin Operation Using Example RS-485 Bus Transceiver.

3.5.2. RS485

Es wird ein 4-Draht RS485-Bus verwendet, damit alle MCP39F501 die Informationen bzw. Befehle empfangen können. Dennoch antwortet nur der IC, welcher mit der gesendeten Adresse übereinstimmt.
-> RPI ist Master, Slave sind MCP39F501

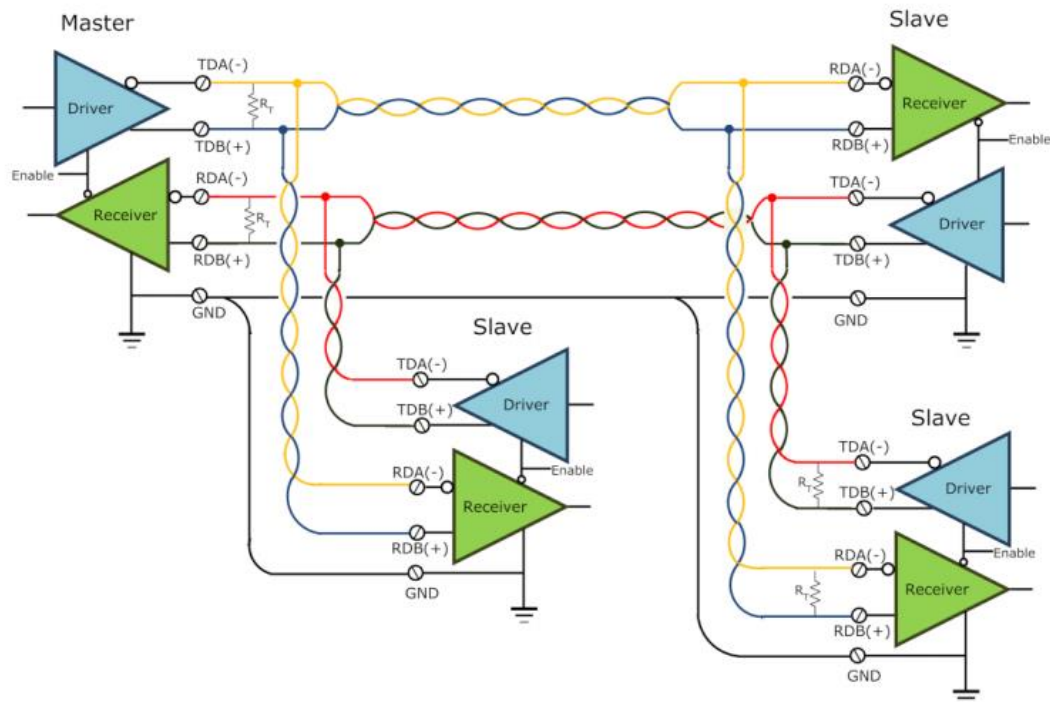


Figure 18: RS-485 Four-Wire Configuration

Bei der Kommunikation sind die Ausgangspegel/Eingangspegel zu beachten!
 -> Gefahr für Bauteilzerstörung!

3.6. Versorgungsplatine

Aufgabe der Versorgungsplatine ist es, die +12V Eingangsspannung auf +5V, +3,3V umzuwandeln. Zur Wandlung wurde ein DC/DC-Wandler verwendet, um die Verlustleistung zu minimieren. Verwendet wurde ein 7805 und 7833.

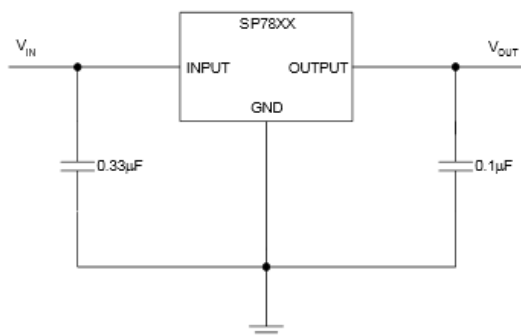
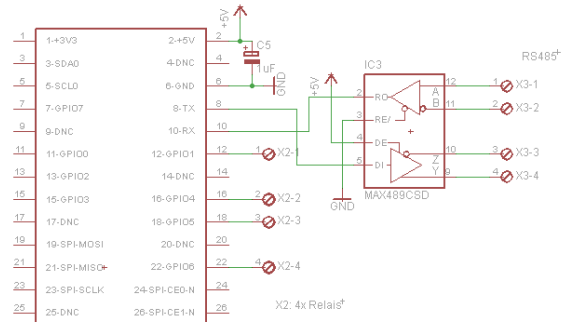


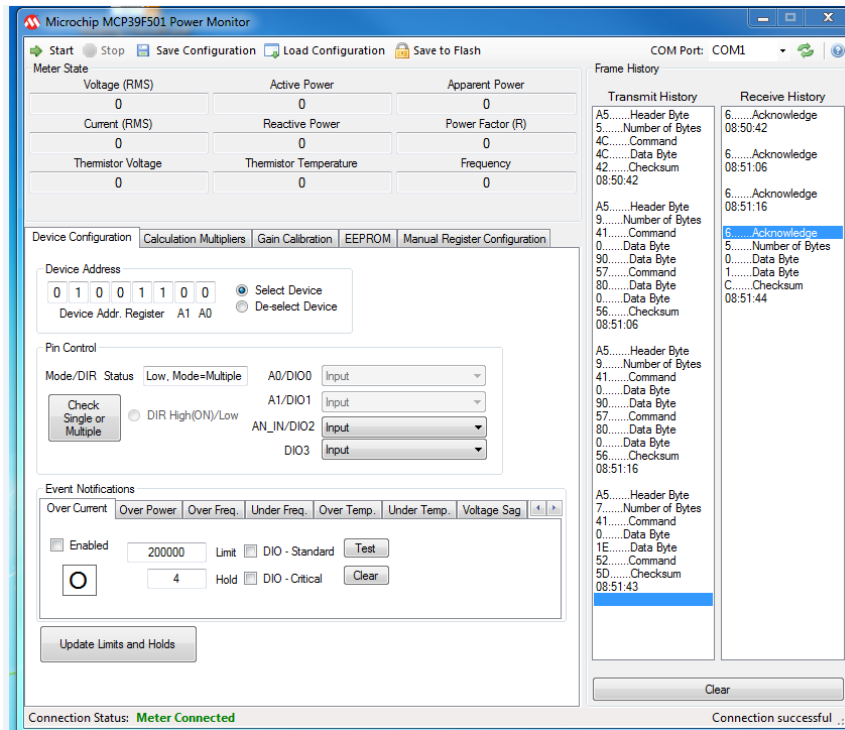
Figure 8. Fixed Output Regulator of SP78XX



10

3.7. Programmierung des MCP39F501

Zur Einstellung des Range, Gain, ... wurde die Software „MCP39F501 Power Monitor Utility“ verwendet.



In den Registern GainCurrentRMS/GainVotlageRMS und Range kann man die Faktoren einstellen, um das Ergebnis zu korrigieren.

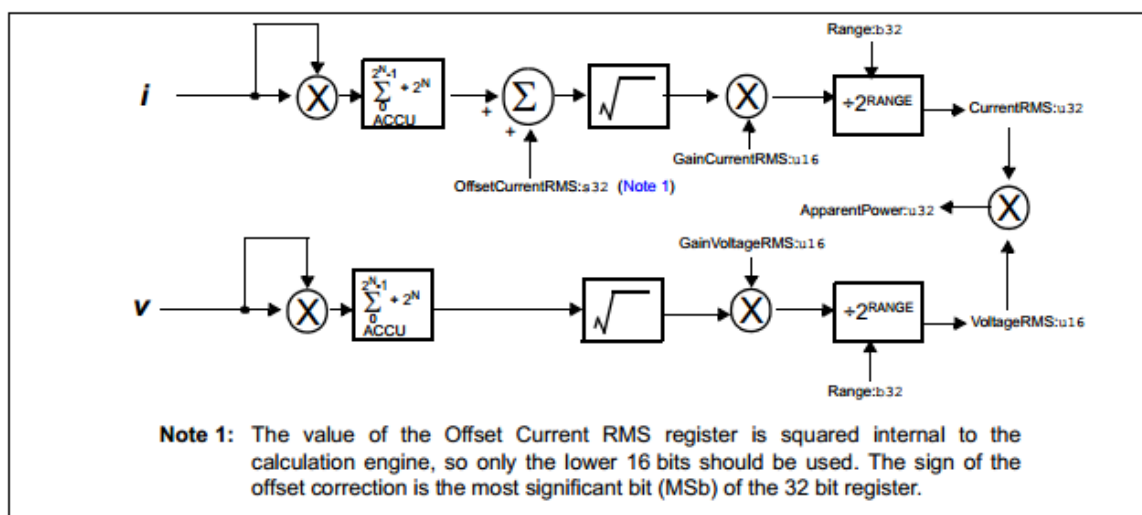


FIGURE 5-2: RMS Current and Voltage Calculation Signal Flow.

→ **Gählter Wert für GainVoltageRMS: 6DD**

Device Configuration					Calculation Multipliers	Gain Calibration	EEPROM	Manual Register Configuration
Register Address	Registers	Attribute	Register Length (Bits)	Value (Hex)				
0x0008	VoltageRMS	R	16	0145				
0x000A	ActivePower	R	32	0000393C				
0x000E	ReactivePower	R	32	0BFD73C2				
0x0012	ApparentPower	R	32	00000000				
0x0016	PowerFactor	R	16	7FFF				
0x0018	LineFrequency	R	16	C220				
0x001A	ThermistorVoltage	R	16	0154				
0x001C	EventFlags	R	16	0000				
0x001E	SystemStatus	R	16	0001				
0x0020	Reserved	R	16	0000				
0x0022	Reserved	R	16	0000				
0x0024	Reserved	R	16	1234				
0x0026	SSIDDeviceAddress	R/W	16	004C				
0x0028	GainCurrentRMS	R/W	16	0001				
0x002A	GainVoltageRMS	R/W	16	6dd				
0x002C	GainActivePower	R/W	16	0064				
0x002E	GainReactivePower	R/W	16	7392				
0x0030	OffsetCurrentRMS	R/W	32	00000006				
0x0034	OffsetActivePower	R/W	32	00000000				
0x0038	OffsetReactivePower	R/W	32	00000000				
0x003C	DCOffsetCurrent	R/W	16	0000				
0x003E	PhaseCompensation	R/W	16	00F7				
0x0040	AppPowerDivisorDigits	R/W	16	0003				
0x0042	SYSTEMConfiguration	R/W	32	3F000000				
0x0046	DOUTConfiguration	R/W	16	049B				
0x0048	Range	R/W	32	0000000A				

→ **Gewählter Wert für Range: A**

Device Configuration					Calculation Multipliers	Gain Calibration	EEPROM	Manual Register Configuration
Register Address	Registers	Attribute	Register Length (Bits)	Value (Hex)				
0x0030	OffsetCurrentRMS	R/W	32	00000006				
0x0034	OffsetActivePower	R/W	32	00000000				
0x0038	OffsetReactivePower	R/W	32	00000000				
0x003C	DCOffsetCurrent	R/W	16	0000				
0x003E	PhaseCompensation	R/W	16	00F7				
0x0040	AppPowerDivisorDigits	R/W	16	0003				
0x0042	SYSTEMConfiguration	R/W	32	3F000000				
0x0046	DOUTConfiguration	R/W	16	049B				
0x0048	Range	R/W	32	0000000A				
0x004C	CalibrationCurrent	R/W	32	0000C350				
0x0050	CalibrationVoltage	R/W	16	0898				
0x0052	CalibrationActivePower	R/W	32	0001ADB0				
0x0056	CalibrationReactivePower	R/W	32	0001741F				
0x005A	AccumulationIntervalParameter	R/W	16	0002				
0x005C	Reserved	R/W	16	0000				
0x005E	OverCurrentLimit	R/W	32	00030D40				
0x0062	Reserved	R/W	16	0000				
0x0064	OverPowerLimit	R/W	32	0006B6C0				
0x0068	Reserved	R/W	16	0000				
0x006A	OverFrequencyLimit	R/W	16	CF08				
0x006C	UnderFrequencyLimit	R/W	16	B880				
0x006E	OverTemperatureLimit	R/W	16	01A4				
0x0070	UnderTemperatureLimit	R/W	16	0096				
0x0072	VoltageSagLimit	R/W	16	0320				
0x0074	VoltageSuroeLimit	R/W	16	0960				

Als Last wurde eine Herdplatte verwendet (1800W).

3.7.1. Protokoll

z.B.:

Frame History	
Transmit History	Receive History
A5.....Header Byte	6.....Acknowledge
5.....Number of Bytes	08:50:42
4C.....Command	
4C.....Data Byte	6.....Acknowledge
42.....Checksum	08:51:06
08:50:42	
A5.....Header Byte	6.....Acknowledge
9.....Number of Bytes	08:51:16
41.....Command	6.....Acknowledge
0.....Data Byte	5.....Number of Bytes
90.....Data Byte	0.....Data Byte
57.....Command	1.....Data Byte
80.....Data Byte	C.....Checksum
0.....Data Byte	08:51:44
56.....Checksum	
08:51:06	
A5.....Header Byte	
9.....Number of Bytes	
41.....Command	
0.....Data Byte	
90.....Data Byte	
57.....Command	
80.....Data Byte	
0.....Data Byte	
56.....Checksum	
08:51:16	
A5.....Header Byte	
7.....Number of Bytes	
41.....Command	
0.....Data Byte	
1E.....Data Byte	
52.....Command	
5D.....Checksum	
08:51:43	
Clear	
Connection successful	

Weitere Informationen befinden sich im Datenblatt.

3.8. Bauteile

Name	Wert	Stück
C1-4	33n	6
C5	0,01uF/ 10n	3
R1	240k	1
R2	4k7	1
R3,4,8,9	1k	4
R5	Shunt 0,025R	1
R6,7	500k	2
R10	10k	1
R11	120R	6
T1	BC517	1
D1	1N4148	1
IC	MCP39F501	1
IC	MAX489CSD	2
IC	7805	1
IC	7833	1
L	Ferrit bead 0805 Automotiv 150R R15	5
Relais	Monostabiles Relais 2-polig	1

4. Software

4.1. Betriebssystem und installierte Programme/Bibliotheken

Das installierte Betriebssystem ist Raspbian, welches auf die Linux Distribution Debian Wheezy aufbaut.

4.1.1. Installierte Programme und Bibliotheken

- Apache 2 (v. 2.2.22)
Webserver für das Webinterface
- PHP und PHP-CLI(v. 5.4.35)
PHP Interpreter mit Kommandozeilen-Interface
- php5-sqlite
PHP-Erweiterung um mit SQLite zu kommunizieren
- SQLite3 (v. 3.7.13)
Einfache, sehr Ressourcen-schonende Datenbank
- PEAR und PEAR-Packages

4.1.2. Verwendete, jedoch nicht installierte Bibliotheken

PHP:

- PhpSerial
Vereinfacht die Benutzung einer Seriellen Schnittstelle in PHP
- System_Daemon
Ermöglicht das starten eines PHP-CLI Programms als Daemon (Service)
(installiert über PEAR)

Javascript:

- Rickshaw
Eine Bibliothek zum einfachen Zeichnen von Graphen
- D3
Grafik-Bibliothek, auf die Rickshaw aufgebaut ist
- JQuery
Bibliothek zum einfachen und schnellen Bearbeiten des Webseiten-Inhalts
- JQuery UI
Moderne und einfach benutzbare Benutzeroberflächen-Werkzeuge
- MenuMaker
Zum schnellen Erstellen von modernen Menüleisten

4.2. Funktionsprinzip

Grundsätzlich besteht das Programm aus 2 Teilen:

- Der SmartPlug Service ist ein im Hintergrund laufendes Programm, das den Großteil der Funktionalität bereitstellt.
Die Konfiguration wird in eine Datei geschrieben, wobei diese jederzeit geändert werden kann und automatisch übernommen wird.
Zum Prüfen, ob eine Steckdose ein- oder ausgeschaltet werden soll wird ein Shell-Script aufgerufen, das bei Bedarf vom Benutzer ohne Neustart des Services bearbeitet werden kann.
- Das Webinterface bildet die Schnittstelle zum Benutzer. Es zeichnet den Stromverbrauch als

Graphen und ermöglicht das Konfigurieren über eine graphische Oberfläche.

4.2.1. Blockschaltbild

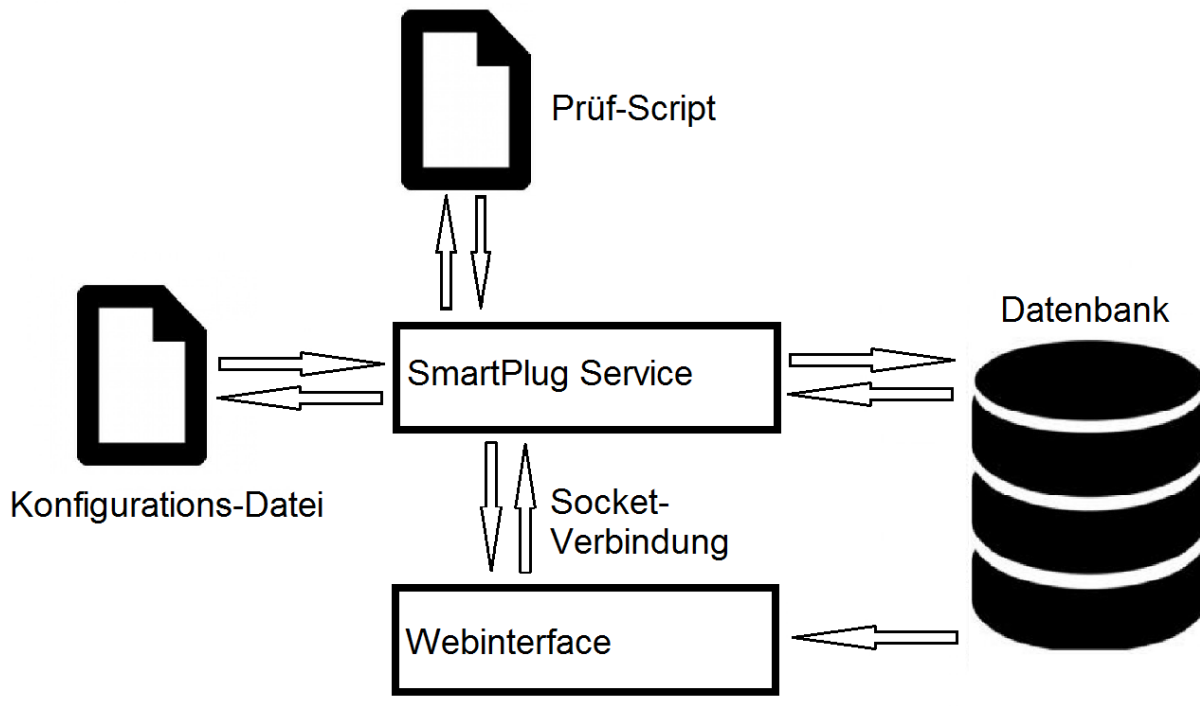


Abbildung 1 Blockschaltbild Programme

4.3. SmartPlug Service

Nach dem Start des Services wird die run() – Funktion der Klasse MainLoop gestartet. Diese hat intern eine Endlosschleife, die einmal pro Sekunde den Strom misst und in der Zwischenzeit Socket-Nachrichten behandelt, die gemessenen Werte analysiert und speichert und prüft, ob eine der Steckdosen ein- oder ausgeschaltet werden soll.

Der Service besteht hauptsächlich aus folgenden Klassen:

- **DBManager**
Ist für alle Datenbank-Operationen zuständig.
- **Log**
Schreibt Nachrichten mit weiteren Details, wie Zeit und aufrufender Klasse, versehen in die Log-Datei.
- **AllRelaisConfigurations**
Enthält die Konfiguration für alle vier Steckdosen / das Verhalten aller Relais. Die Klasse liest die Konfigurations-Datei ein und schreibt Änderungen hinein.
- **RelaisConfiguration**
Enthält die Konfiguration für eine Steckdose / ein Relais.
- **RelaisController**
Startet den Prüf-Script (abhängig von der Konfiguration) und wertet die Ergebnisse aus.
- **MeasureValue**
Speichert einen Messwert. Es wird das letzte Minimum, Maximum, der Durchschnitt und der letzte Wert gespeichert. Außerdem wird eruiert, ob die Wertänderung seit der letzten Abfrage

innerhalb einer definierten Toleranz ist.

- **MeasureDevice**
Enthält je eine Instanz von MeasureValue für jeden Wert, der gemessen wird.
- **Measurement**
Ist für die Kommunikation mit dem Strommess-Chip zuständig.
- **Serial**
Ist eine Wrapper-Klasse für die PhpSerial – Bibliothek.
- **Socket**
Beinhaltet einen Server-Socket und wertet die Nachrichten nach einem eigenen Protokoll aus.
- **GPIO**
Erleichtert das Schreiben auf und das Auslesen von GPIOs.

4.4. Datenbank

Die Datenbank besteht aus drei identischen Tabellen:

- **DayMeasurements**
Hier werden die Messwerte im 5-Minuten-Takt gespeichert.
- **MonthMeasurements**
Hier werden die Messwerte im 1-Stunden-Takt gespeichert.
- **YearMeasurements**
Hier werden die Messwerte einmal pro Tag gespeichert.

DayMeasurements	MonthMeasurements	YearMeasurements
device INT	device INT	device INT
datetime INT	datetime INT	datetime INT
active FLOAT	active FLOAT	active FLOAT
activeMin FLOAT	activeMin FLOAT	activeMin FLOAT
activeMax FLOAT	activeMax FLOAT	activeMax FLOAT
reactive FLOAT	reactive FLOAT	reactive FLOAT
reactiveMin FLOAT	reactiveMin FLOAT	reactiveMin FLOAT
reactiveMax FLOAT	reactiveMax FLOAT	reactiveMax FLOAT

4.5. Webinterface

Das Webinterface zeichnet Graphen für die Tages-, Monats- und Jahresmesswerte für jedes Gerät einzeln.

Außerdem ist es möglich das Schaltverhalten der Relais per grafischer Oberfläche zu konfigurieren. Die Funktionalität bieten folgende PHP-Klassen:

- CMS
Ist für das Erstellen des HTML-Headers und Bodys zuständig.
- Chart
Bindet die JS-Bibliothek Rickshaw ein.
- DBManager
Ist die Webserver-seitige Anbindung an die Datenbank.
- SocketClient
Bildet das Gegenstück der Klasse Socket im SmartPlug Service.

Es gibt folgende Seiten:

- index.php zeigt die Graphen.
- settings.php bietet die Funktionalität mit der Konfiguration.
Um diese Seite aufzurufen, muss der Service laufen, da die Konfiguration über die Socket-Verbindung abgefragt wird.
- data.php ist eine Seite, die nur für den Gebrauch mit AJAX gedacht ist und die Messwerte im JSON-Format ausgibt.

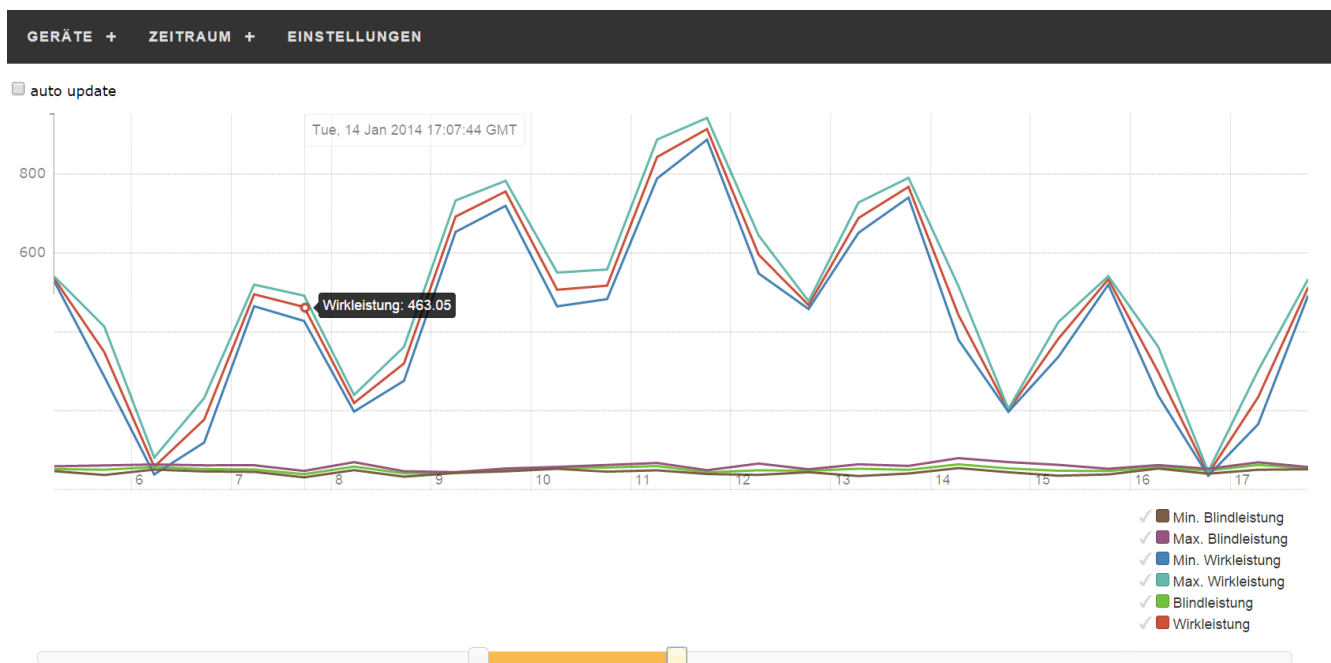


Abbildung 2: index.php

GERÄTE + ZEITRAUM + EINSTELLUNGEN	
Relais Default:	
Überprüfungs-Script:	<input type="text" value="defaultcontrol.sh"/>
Ausgangswert überschreiben:	<input type="text" value="-1"/>
Parameter:	<input type="text" value="t >= 0"/>
	<input type="text" value="t <= 1440"/>
	<input data-bbox="715 589 738 611" type="button" value="+"/>
Relais R1:	
Überprüfungs-Script:	<input type="text" value="defaultcontrol.sh"/>
Ausgangswert überschreiben:	<input type="text" value="-1"/>
Parameter:	<input type="text" value="t>=0"/>
	<input type="text" value="t<=144"/>
	<input data-bbox="715 813 738 835" type="button" value="+"/>
Relais R2:	
Überprüfungs-Script:	<input type="text" value="defaultcontrol.sh"/>
Ausgangswert überschreiben:	<input type="text" value="-1"/>
Parameter:	<input type="text" value="t>=0"/>
	<input type="text" value="t<=1440"/>
	<input data-bbox="715 1037 738 1059" type="button" value="+"/>
Relais R3:	
Überprüfungs-Script:	<input type="text" value="defaultcontrol.sh"/>
Ausgangswert überschreiben:	<input type="text" value="-1"/>
Parameter:	<input type="text" value="t>=0"/>
	<input type="text" value="t<=1440"/>
	<input data-bbox="715 1261 738 1283" type="button" value="+"/>
Relais R4:	
Überprüfungs-Script:	<input type="text" value="defaultcontrol.sh"/>
Ausgangswert überschreiben:	<input type="text" value="-1"/>
Parameter:	<input type="text" value="t>=0"/>
	<input type="text" value="t<=144"/>
	<input type="text" value="t>=0"/>
	<input data-bbox="715 1529 738 1552" type="button" value="+"/>
<input type="button" value="Speichern"/>	

Abbildung 3: settings.php