SunZilla Score v1.0

Smart Control for Renewable Energies

Construction Manual

Authors: Laurin Vierrath, Lea Schmid

Publisher: SunZilla UG (haftungsbeschränkt)

January 2017

Preamble

This Project was funded by the POC21 Patreon Funds. This fund has been set up after POC21, a five-week work camp that brought together designers, scientists, engineers and business wizards to advance 12 open source hardware projects.

If you like open source and / or this projects, please make your contribution to the POC21 Patreon Community. More details:

POC 21

POC21 Patreon Community

Licence

Acknowledgement

Thanks to everybody who helped to realize and finalize this project and this documentation.

Special thanks to:

POC21 Patreon group, August & Piela Konstruktiv GbR, Steffen Eyhorn, Martin Jäger, Felix Obée

Disclaimer

This project comes "`as is"'. The publisher does not take any responsibility for damages or harms caused to equipment, humans or animals! Please be aware that this manual allows you to build and operate electric facilities which can be dangerous and/or cause damages and harms.

Please read the documentary entirely before you get started.

Preamble2
Licence2
Acknowledgement2
Disclaimer2
Introduction4
SunZilla4
SunZilla score4
Hardware6
Get the PCB6
Get the Electric Components6
Soldering6
Starting the Board6
A detailed look at some aspects8
Software8
Programming the ARDUINO8
Programming the ESP82669
Must Know:11
Electro technical Background: Switching Currents12
Safety Instructions13
Fusing13
Good to know14
Communication between ESP8266 and ARDUINO14
Measurement accuracy14
Critical Points and Improvements15
PV-Channel: MOSFET configuration15
Trace Thickness15
3.3V Power Supply15
Current Measurement15
Consistent labelling15

Introduction

SunZilla

SunZilla is an open source, modular solar generator to-go. It is designed to replace diesel generators in areas where people don't have access to a reliable energy grid as in remote areas, music festivals, sport events or camping. It enables anybody to generate, store and use clean electricity in any place in the world. It is designed to be set up within five minutes by anybody and there are no special skills or tools needed.

SunZilla score

SunZilla score is a product which allows the user of SunZilla to control the solar generator. It enables the user to check the battery charge state, set preferences and control the system via a WiFi-interface. Also it logs data of all sensors and can be used for system performance evaluation and diagnostics. The aim of the project is to find a management and monitoring solution which is independent from manufactures. Whatever equipment you have or buy, this system will work with it.

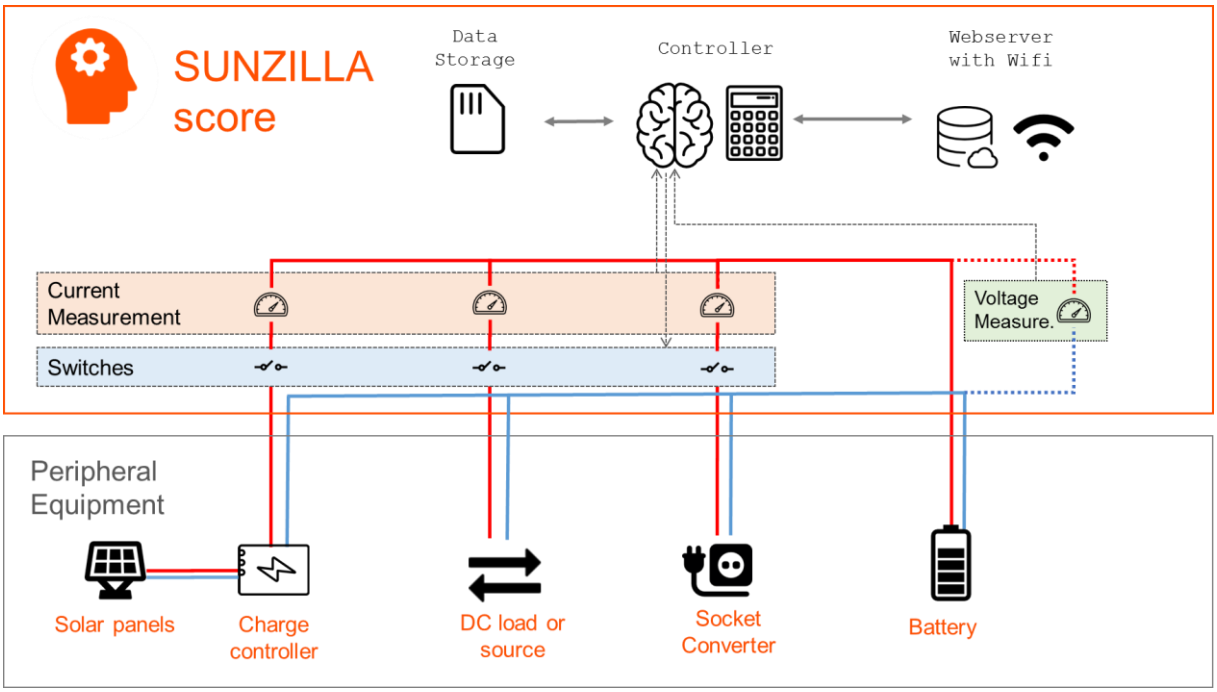
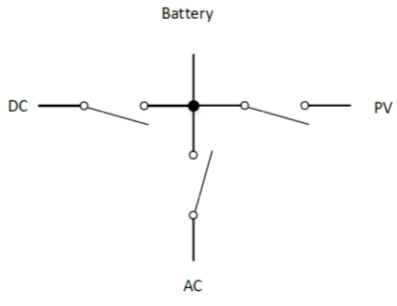


Abbildung 1: Integration of SunZilla score in (existing) off grid power plants.

Key features of SunZilla score

Basic features

- **4 Channel Energy Junction**, each channel equipped with power flow measurement and switch, dedicated to:
  - 1 Battery
  - 1 Solar Charger
  - 1 Load
  - 1 Universal Channel (DC Load or Energy Source)
- **Voltage levels**: 12…24 V
- **Safety Features**
  - Load Cut off when
    - Low voltage / low battery charge state



        - Current exceed
    - Over/under voltage control
- **Monitoring / Data Logging**
    - All System data, including power flows, energy consumption and switch states are stored on a microSD-card. The data is stored as ASCII file and can easily be loaded in the Excel, Matlab a.s.o
    - SunZilla Score is able to upload live data to a central server (based on php/mysql)
- **M2M (Machine to Machine) / IoT (Internet of Things) Interface**
    - All Switch Functions, data access is possible through Laptop / Smartphone via WiFi

Advanced Features

From the basic features above advanced features evolve, which will be implemented subsequently and on-demand by the SunZilla developers and community.

- Defining user profiles for switching events, such as
    - Load 1 should be on if powered through the sun
    - Load 2 shall only be switched on between 2 and 3 pm
    - …
- Energy trading with neighbours
- Energy saving challenges
- A native App for Smartphones (Integration of all Option and data visualisation) will be released in 2017

Technical Datasheet

ABSOLUT MAXIMUM VOLTAGES			**ABSOLUT MAXIMUM CURRENTS**		
VOLTAGE DIVIDER FOR MEASUREMENT*	29.7	V	MAXIMUM CURRENT (PCB WITH 70µM COPPER LAYER)*	20	A
…VOLTAGE CONVERTER	32	V	Plug/socket	25	A
…MOSFET DRIVER	40	V	Current Measurement IC	25	A
…MOSFET SWITCH	60	V	MOSFET Switch	28	A
VOLTAGE LEVELK	**12/24**	**V**	Own consumption @ 12 V	0,5	W
DIMENSIONS	80x100	mm²	Own Consumption @ 24 V	0,8	W

* (limitational component)

Above listed you find the specifications for the voltages and currents. The smallest value is the limiting value. If you want to increase these values, you have to change the part with the smallest value first. The other parts are denoted to know their limits.

Hardware

Get the PCB

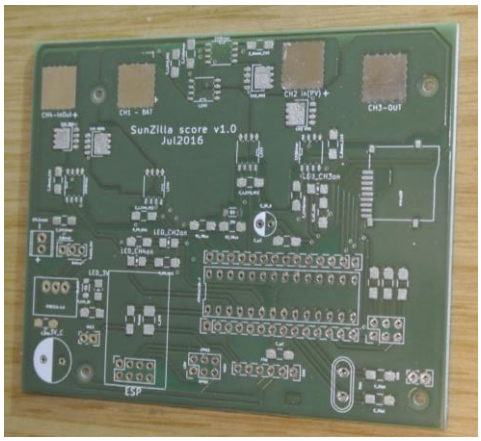


Abbildung 3 Blank SunZilla PCB

The SunZilla score PCB (Printed Circuit Board) was designed using the open source software KiCad EDA. Please visit SunZillas GitHub repository to download the KiCad-Files.

There is two options to acquire the PCB:

- Order your PCB / PCBs at PCB manufacturer of your choice. We used multi-circuit-boards.eu.
- Order PCB from SunZilla online-shop (future)

Get the Electric Components

See Bill of Materials and Indication.pdf to find all the parts that you will need. Order at your local supplier or deliverer. If you do not know where to start, check out: Farnell.com, arrow.com, digikey.com, mouser.com, rs-online.com.

Soldering

As a soldering order we recommend the general rule "From small to big" – first, the small parts, then the bigger parts. Remember that the pins connected to GND need more heat, because the GND pin is quite big and all the entire pin must be heated up. We used about 400° C.

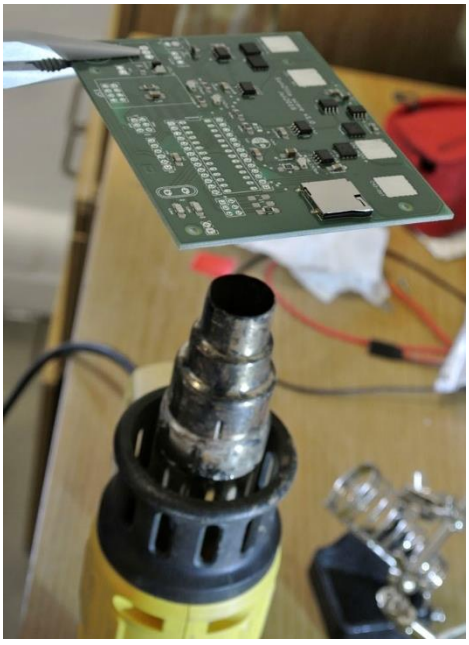


Abbildung 4 Soldering with a Heat Gun

Most of the parts can be hand soldered. We recommend a soldering iron of at least 80 W and a thin (0.8mm) tip. Only the MOSFETs cannot be hand soldered. They must be attached at first. We used soldering paste and a heat gun – be careful however, as excessive heat will result in partly or full failure of parts, which can cause weird, almost impossible-to-debug problems. The other SMD parts can also be soldered in that process or eventually by hand – as you like. Common open-source pizza-oven-reflow-soldering-setups are preferrable, but not necessarily the only solution.

Videos explaining to solder with solder paste and heat gun can be found on the internet, for example that one: https://www.youtube.com/watch?v=0XENpPtisnM

Please make sure that you heat up the PCB equally at all side to reduce tensions in the material.

We also tried using a soldering mask but we found that it is easier to just attach the solder paste by hand

Note: Language in this manual…

Is not always consistent. Please be sure to know what's the same for us…

Channel 2	PV Channel
Channel 3	AC channel
Channel 4	DC Channel
Arduino	ATmega328

Testing the measurement and switches

After soldering the mandatory parts and BEFORE plugging the Microcontrollers (both off them), you can start the score by powering it and execute this little test routine.

Powering the Board

The Board can be powered by two different ways, which can be selected with a 1x2 jumper on the *Supply_Sel* Pin Headers. If you put the jumper on the Battery-indicated ports, the board will be powered by the source that is connected to the 24V-Mounting tab (which will usually be a battery). If you put the jumper to the other port, SunZilla score is fed by a power supply connected to the Pins *EXT_Supply*.

External ON-OFF Switch

In a normal operation you want to be able to shut down the board entirely. This is why you should think of replacing the jumper by the two pins of an external hardware switch. If the board is not powered, all Channels will be off including the Microcontroller (ATMega328) and the WiFi-Shield (ESP8266). The only standby power is drawn by the voltage divider for the voltage measurement (16 kOhm + 2kOhm). The standby losses thus are P_loss = U^2 / R = 24² V / 28 kOhm = 32 mW.

Test routine – Please find and open SunZilla_score_v1.0_testroutine.pdf

1. Power the board. LED_3V should be on.
2. The voltage between Ground and current Measurement Pins should be 1.65 V
3. The Voltage between Ground and Voltage Measurement should be 1/9 of the voltage between Battery and GND (possibly your supply voltage).
4. Power the Mosfet Pins with 3.3 V (not more!), for example by connecting them with +. The LEDs LED_CH2on, LED_CH3on and LED_CH4on should be turned successively on.

If one test fails, check if the soldering process was executed correctly. Step 4 does only check the LEDs and actually not the function of the Mosfet Drivers or the mosfet switches!

A detailed look at some aspects

the ATMega328p pin setting:

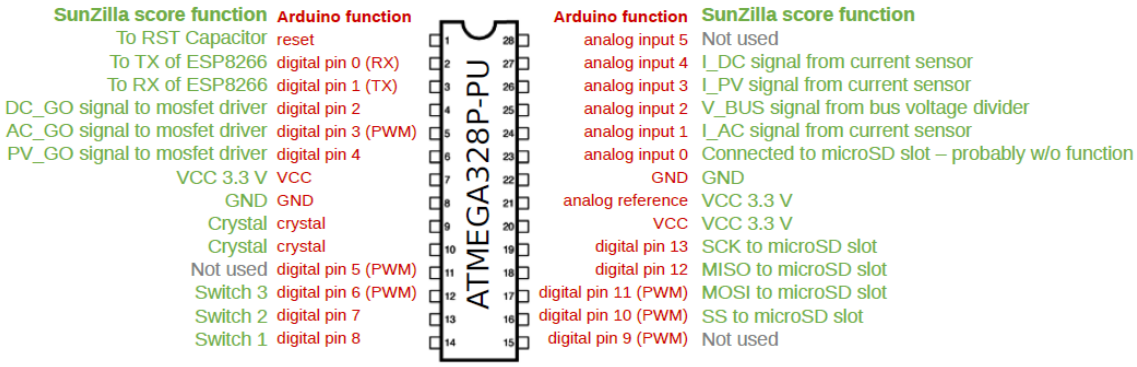


Abbildung 5 Pin Setting of ATMega328p-ou

Why are there two MOSFETs in series?

A MOSFET carries dependent on the construction a reverse body diode. That means, it is not possible to switch currents going from Source to drain!

If currents in both directions shall be switched it is necessary to apply two antiseries MOSFETs.

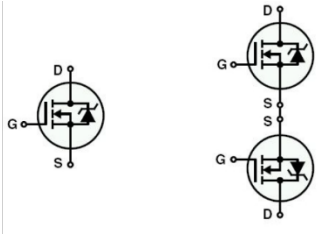


Abbildung 6 Why 2 Mosfets?

Software

The software can be loaded to both chips through an FTDI / CP2102 which can be purchased on the internet for some dollars – make sure to buy a version that can handle both 5V and 3.3V.

Remember one important thing: The FTDI feeds 5 V to the chip. This will damage the ESP8266! Be sure to not connect the voltage supply of the FTDI with the voltage supply of the chips by moving the voltage supply pin out to the socket or by removing the jumper indicated in photo.

Figure 7 FTDI with removable PIN for Power Supply

Remark: Both the Arduino and the ESP8266 will be programmed through the Arduino IDE. We had big performance problems compiling and uploading software on Windows Computers. On Linux those problems did not occur, because Linux is awesome and superior and we totally want to start a Linux-vs-Mac-vs-Windows-bitchfight here ;)

Programming the ATMega328P-PU

The microcontroller ATMEGA328P-PU is built as *Arduino on a Breadboard*, it just behaves like an Arduino Pro. It is possible and recommended to use the *Arduino IDE* to program it. Before you can program the ATMega823 using the Arduino IDE, you need to initialize the chip by Installing the Arduino Bootloader. If the Boot Loader is already burned to the chip, you can skip the next chapter.

Important notice: Since the microcontrollers supply voltage is only 3.3 V, its frequency is only (roughly) 8 MHz. Make sure to select to corresponding board in your Arduino IDE Software (e.g. *Tools -> Board -> `Arduino Pro or Pro mini (3.3V, 8Mhz) with ATmega328'*) for the bootloader and for programming the chip!

Installing the bootloader

To use the ATMEGA328P-PU as an Arduino we must upload the right bootloader to the ATMEGA. To do so you can for example follow the instructions on https://www.arduino.cc/en/Tutorial/ArduinoISP. Don't forget to connect a 10 µF capacitor between RESET and GROUND on the board you are using as programmer.

Uploading Software to the ATMega328

Set in the FTDI according to photo. Remember to remove the ESP8266 when programming the Arduino (left red circle) and to feed in supply voltage to the score Board.

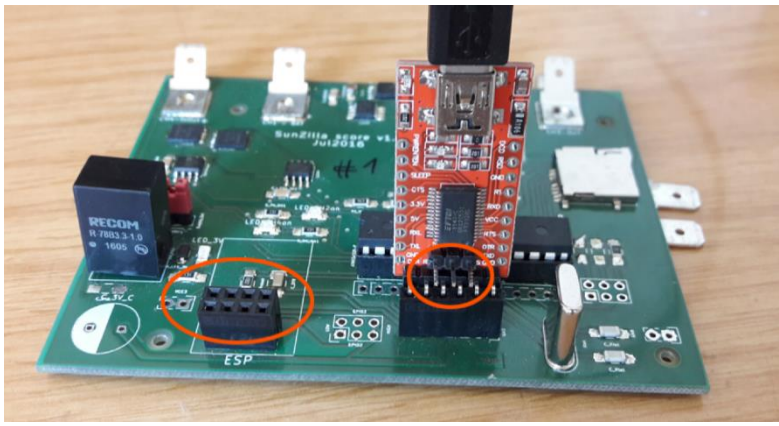


Abbildung 8 SunZilly score with FTDI and without ESP

Important notice: Most FTDI-Interfaces deliver only 5V to the board which would instantly destroy the chip. There is FTDI-Interfaces which can deliver 3.3V and 5V, which can be selected by a Jumper (right red circle). We recommend to remove the Jumper because 1. You might confuse the voltages and feed in 5V and 2. There might not be enough power to feed the entire score Board

Programming the ESP8266

HARDWARE NEEDED

We made a quick setup on a perfboard to be able to program the ESP. You can see it in the figure below. Spending those extra 10 minutes soldering a nice ESP-programmer might be worth your valuable time.

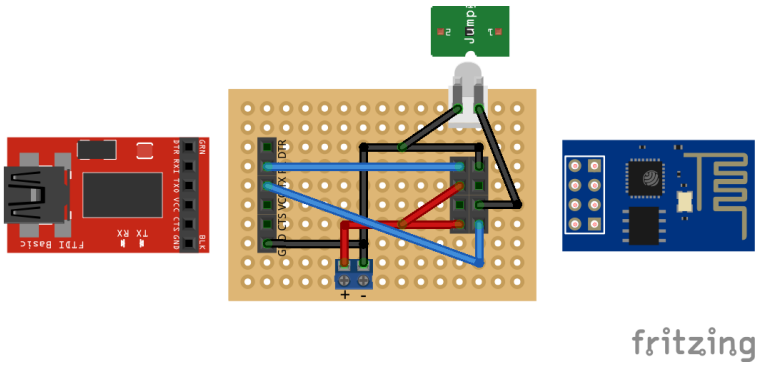


Abbildung 9Perfboard to program the ESP

The voltage supply by the FTDI might not be powerful enough to feed the ESP! Make sure you use an external 3.3 V supply. The ESP8266 must be connected in the position you can see it in on the right hand side of the perfboard.

ESP8266	FTDI	Power Supply 3.3V
CH.PD		+
VCC		+
GND		-
TX	RX	
RX	TX	
GND	GND	

Connections on the Board

The Jumper is to select the start mode of the ESP8266, the mode the ESP enters when you put the chip in or when you restart th power supply.

Jumper set	Yes	No
At Startup (switching on the supply voltage or putting in the chip)	Programming Mode	Normal Mode = Executing Code
After Programming	Normal Mode = Executing Code	-

If you are in normal mode and want to program the chip, you must interrupt the voltage supply.

SOFWARE PREPERATION

Before you can program the ESP with the Arduino IDE, you need to download the chip information as described in the next chapter.

To program the ESP8266 we use Arduino IDE, which must be prepared to communicate with ESPs. To do so open your Arduino IDE and Click *File -> Preference* and enter "http://arduino.esp8266.com/stable/package_esp8266com_index.json" to *Additional Boards Manager URLs*.

The next step is to click *Tools -> Board:".." -> Board Manager*. Scroll down until you see esp8266 and click "install".

Now choose your board being "Generic ESP8266 - Module". You can execute the example to check the memory of your chip. Choose the specifications according to what you found out through the example. We put up the upload speed to .

To check if everything works you can use the example sketch "Hello server" and upload it to your ESP following the steps described earlier.

Acquiring and Uploading the code

Step	Arduino	ESP
	Upload in score, FTDI connected to the designated PINs	**Upload in perfboard as described before**
0	Download the software for your hardware at …	
1	Connecting the FTDI with the USB Port of the computer you want to use to program and the ATMega328 (pin headers designated for FDTI) / ESP (special perfboard)	

2	Open Arduino IDE Selecting right Port by clicking *Tools -> Port*	
3	Open *score_Arduino_vXXX.ino*	Open *score_ESP_vXXX.ino*
4	Select *Tools -> Board -> Arduino Pro or Pro Mini*	Select *Tools -> Board -> Generic ESP8266 Module*
5	Select *Tools -> Processor -> Atmega328 (3.3V, 8MHz)*	Select correct memory size and upload speed
	Click Upload	

Finally: Starting the device

If you're uploads worked correctly, assemble SunZilla score by putting in a FAT-formatted SD-Card and the ESP8266. Start your SunZilla by powering the Board.

Switch to the Operational Manual now!

System Integration

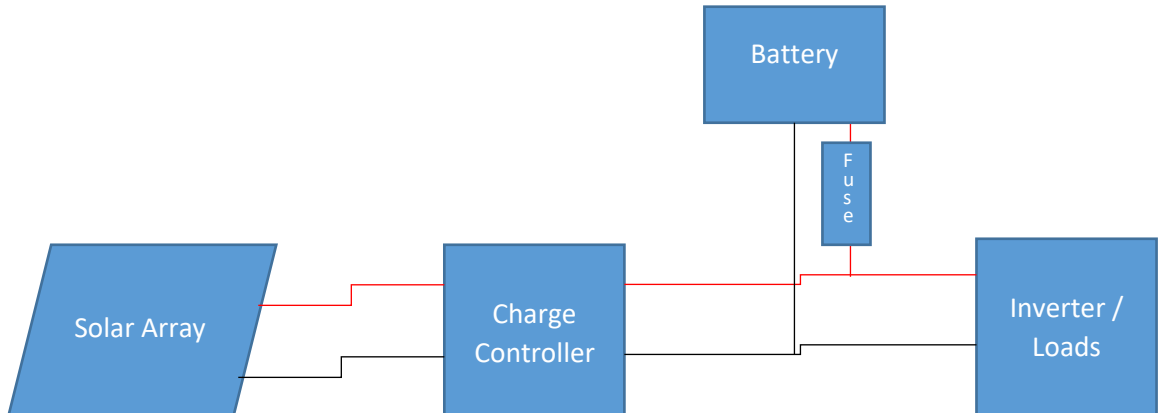


Abbildung 10: Conventional Off grid Power Plants

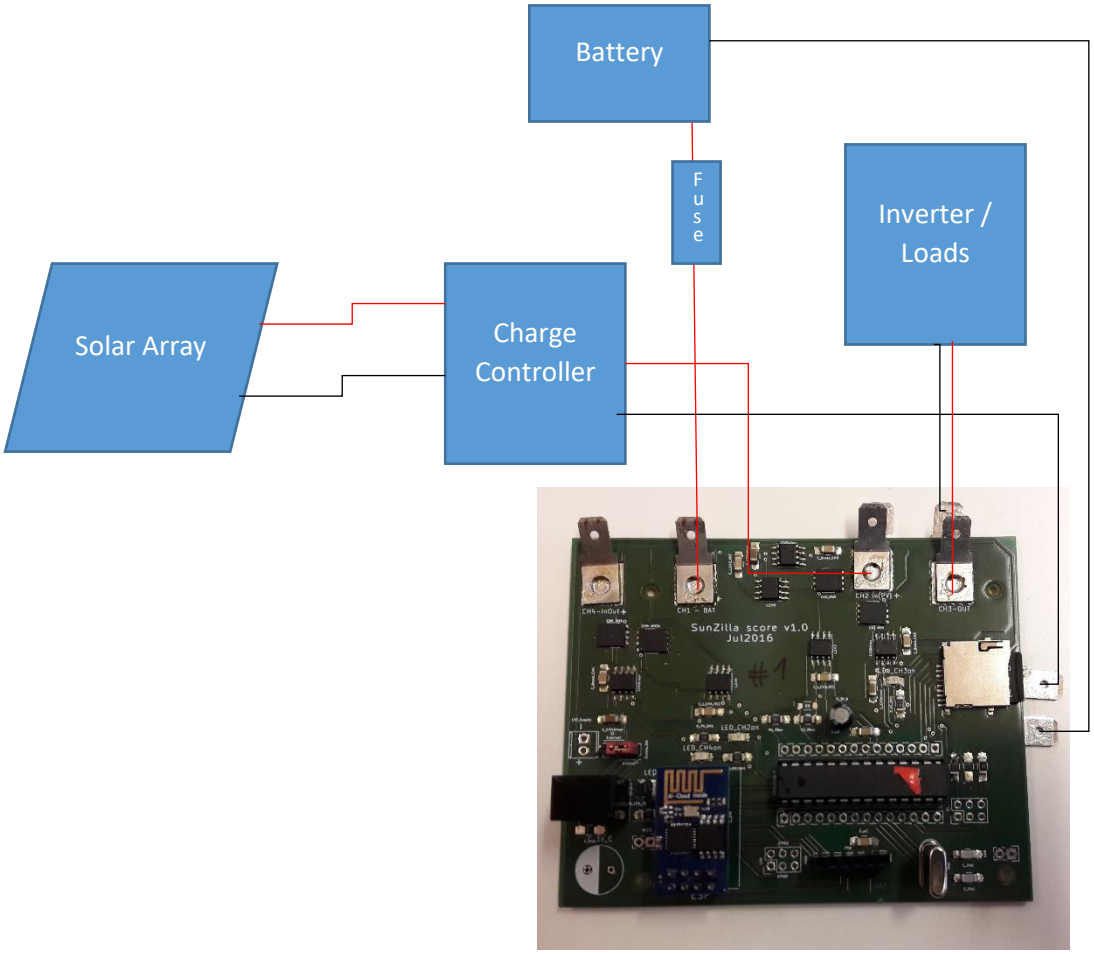


Abbildung 11: Off-grid with SunZilla Score

Must Know:

Electro technical Background: Switching Currents

SunZilla and SunZilla score works with 12/24 V DC. There is one fundamental difference between AC and DC which is important to know and understand SunZilla score.

In AC, the voltage and thus the current have a sinusoidal form including a zero-crossing. This zero-crossing is super important, as it interrupts the power feed. This is why we can disconnect a 110/220 V AC plug from the socket safely by just pulling the plug.

In DC, there is no interruption of the power feed. This is why switching DC currents is much more complicated. If you just pulled a plug carrying DC current from a socket, one runs the risk of creating a current-conducting plasma-channel (like a flash). This plasma channel can damage your hardware, especially the contacts of the respective connector, and as it is very hot might even lead to serious injuries.

So if you want to switch off DC currents you must use equipment that has the power to switch DC currents like a MOSFET. A (standard) relais will not do it (there are expensive DC-relays, we know...)!

This is why it is super important to switch off the channels before you unplug the cables!

Safety Instructions

As we want to control power applications, SunZilla score, used cables, plug and sockets must be appropriate for high currents. On the SunZilla score PCB this is realized by a large width of the current layers. These current traces are highlighted in the Picture.

Any other equipment carrying the high current must be dimensioned to be able to transfer that current.

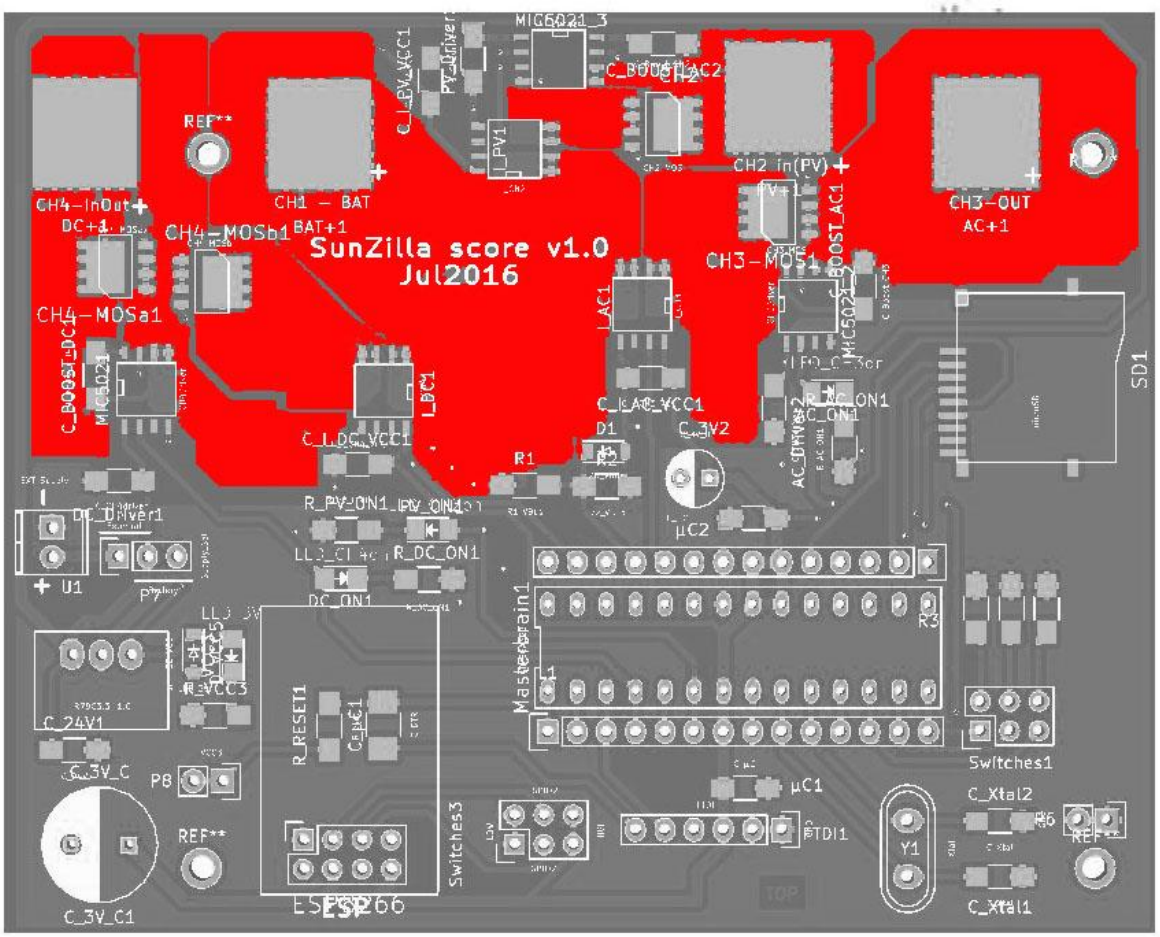


Abbildung 12 Large width for high current transmission parts

Fusing

Energy sources in general need to have a fuse. Though a PV-Charger is an energy source (as it provides energy) it does not necessarily need a fuse. Firstly, many PV chargers come with a fuse. Secondly, the charger hardly can provide more current than the PV arrays nominally does.

The battery definitely needs a fuse. As we allow a maximum current of 20 A, find a fuse of 25-30 A.

Good to know

Communication between ESP8266 and ARDUINO

Main Tasks of the 2 chips are:

ATMega328: Read Currents and Voltage, check for limit exceedance, calculate power flows, safe data to microSD-Card, command MOSFET switches

ESP8266: make the data and switching opportunities available on WiFi

The 2 Chips use Serial Communication at a Baud Rate of 57600 through Pins RX and TX.

This command set is subject to change and extension can be found in command_set.xlsx.

Measurement accuracy

It is important to know that the current measurement has limited accuracy. In this section reasons for this inaccuracy are depicted.

Currents: The measurement interval is -25 A...25 -> A range of 50 A will be mapped on a voltage output interval of 0...3.3 V. The resulting sensitivity is 50 A / 3.3V ~= 15 mA / mV

The Datasheet of the ACS711 current sensors additionally mentions a Total Output Error of 4 %.

Voltage:

The measurement interval is 0 … 30 V -> A range of 30 V is mapped on a voltage interval of 0…3.3V. The resulting sensitivity is 30 V / 3.3 V = 9 mV / mV

Arduino

The Arduino measures in 10-bit resolution: 2^10 = 1024 steps for voltages from 0 to 3.3 V -> A range of 0…3.3 V will be mapped on 1024 steps

Measurement accuracy= 3.3V / 1024 = 3.2 mV / bit = LSB

The Atmega328P-PU has a measurement of +- 2 LSB = +- 6,4mV

That means: The Atmega cannot measure the currents better than +- 96 mA and voltages +- 57 mV.

That means that the power cannot be read more exactly than +- 1 W (12V system) resp. +- 2 W (24 System).

Critical Points and Improvements

PV-Channel: MOSFET configuration

As described earlier, you need 2 antiseries MOSFETs to switch bidirectional appliances. At first sight, a PV charger is a source of energy. It feeds in the power coming in from the PV array. The single MOSFET in the PV channel is able to interrupt the power infeed from the sun.

But when there is no sun the PV charger acts as a load and draws current from the battery to satisfy the standby needs. This might be negligible as it is only some 10 mA. For use cases where the SunZilla unit is permanently in use, this might be the right choice (as a second MOSFET brings more losses). But for use cases that include deactivating and storing the SunZilla units it is recommended to be able to switch off the pv charger entirely. This is why we at SunZilla connect the PV charger to the DC-InOut (and leave the PV in blank). In the next version, the PV channel will be equipped with to MOSFETs.

Long story short: PV Channels needs bidirectional Switch: PV Chargers act as load if solar input is zero. To reduce standby losses and to safely disconnect pv charger must be connected entirely.

Trace Thickness

The power is conducted through the PCB. The Ampacity is given by the width of the trace more than the thickness. The goal was to reach an ampacity of 20 A. This has the be accessed and analysed

3.3V Power Supply

The Recom R78… IC is quite expensive. There is cheaper option to achieve a reliant 3.3 power supply, for example with an MCP16311 plus circuitry. For simplification, the Recom was chosen. In the next version this should be fixed.

Current Measurement

Also for simplification, the quite expensive ACS711 current measurement chip has been chosen. In the next version, this should also be replaced by an OpAmp + Shunt Resistor.

Consistent labelling

Yes, we know we aren't perfect. But we didn't do it to annoy you and we, ourselves were confused by our labelling. There were two different motivations which we were not able to juggle.

First, we wanted to label the channels for their purposes like Battery, PV, AC or DC.

Secondly, we wanted to have the channels labelled more abstract as they serve in general purposes (Channel x)

We will do better the next time, but for now, please use this table to clarify everything:

Channel 1	=	Battery
Channel 2	=	PV
Channel 3	=	AC
Channel 4	=	DC