Inverter Aurora ABB (Power One) Web Monitor (WIM) With Esp8266

by xxreef

Autonomous centraline with an **esp8266** that **grab and store data** from inverter and **show charts** and various **data of production** and can **send notification email** if there are some problems.

It is a quite user-friendly browser based monitoring solution, It's allows to track energy produced on a solar power plant in a simple and intuitive fashion. It's can track key energy metrics as well as the energy produced throughout the lifetime of their solar power plant. ;)

The PCB is free and you can get 10 pcb for 5$ at this link

Why this project :P

I put solar panels over my roof some years agò, the company that installed them had also guaranteed me a production monitoring and analysis system, but they forgot to tell me that it would be free only for the first year, and I would have had to pay to access my data that are stored on a site, the cost is not so enough (70€ for year) but I felt cheated.

Step 1: Demo Video of Multilanguage Interface Served From Esp8266

https://youtu.be/uInRM3YqIv0

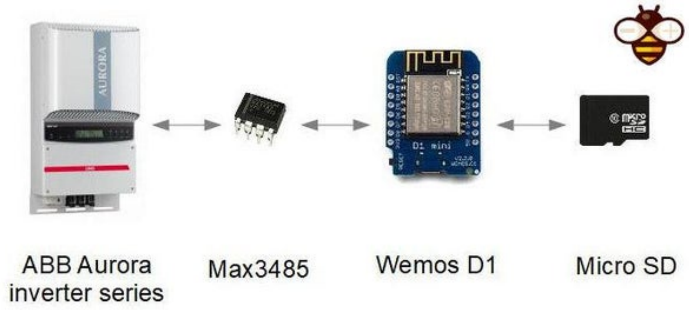
Step 2: Inverter Aurora ABB (ex PowerOne Now Fimer) Supported

Here a partial list of Aurora PV series supported

- PVI-2000
- PVI-2000-OUTD
- PVI-3600
- PVI-3.6-OUTD
- PVI-5000-OUTD
- PVI-6000-OUTD 3-phase interface (3G74)
- PVI-CENTRAL-50 module
- PVI-4.2-OUTD
- PVI-3.6-OUTD
- PVI-3.3-OUTD
- PVI-3.0-OUTD
- PVI-12.5-OUTD
- PVI-10.0-OUTD
- PVI-4.6-I-OUTD
- PVI-3.8-I-OUTD
- PVI-12.0-I-OUTD (output 480 VAC)
- PVI-10.0-I-OUTD (output 480 VAC)
- PVI-12.0-I-OUTD (output 208 VAC)
- PVI-10.0-I-OUTD (output 208 VAC)
- PVI-12.0-I-OUTD (output 380 VAC)
- PVI-10.0-I-OUTD (output 380 VAC)
- PVI-12.0-I-OUTD (output 600 VAC)
- PVI-10.0-I-OUTD (output 600 VAC)"
- PVI-CENTRAL-250
- PVI-10.0-I-OUTD (output 480 VAC current limit 12 A)
- TRIO-27.6-TL-OUTD
- TRIO-20-TL
- UNO-2.0-I
- UNO-2.5-I
- PVI-CENTRAL-350 Liquid Cooled (control board)
- PVI-CENTRAL-350 Liquid Cooled (display board)
- PVI-CENTRAL-350 Liquid Cooled (AC gathering)

Step 3: Project Introduction

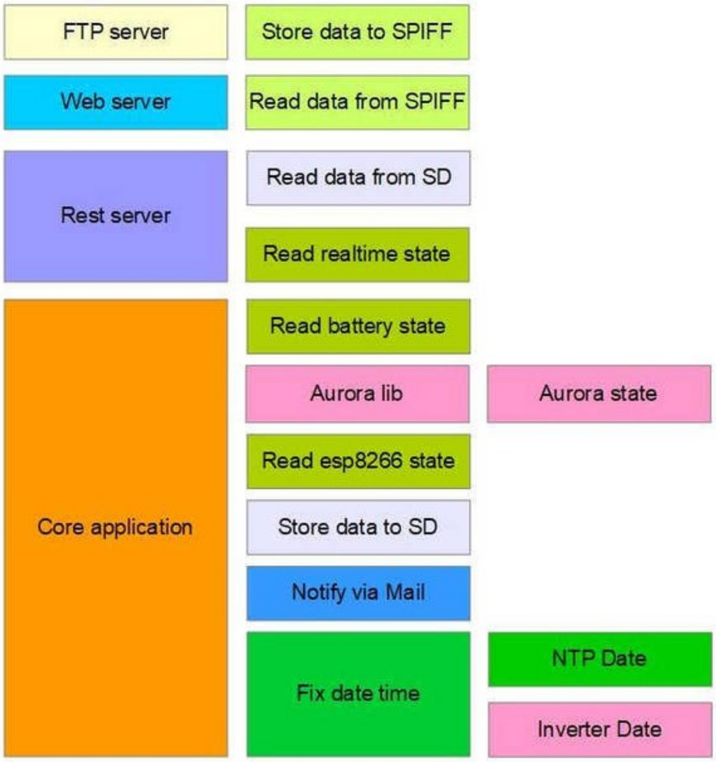
My idea is to use an esp8266 (Wemos D1) with enough power to manage an http server, a rest server and ftp server, naturally with an IC can interface my inverter (ABB Autora - ex PowerOne), all data taken from the inverter will be stored in an SD.



Step 4: Project Layers

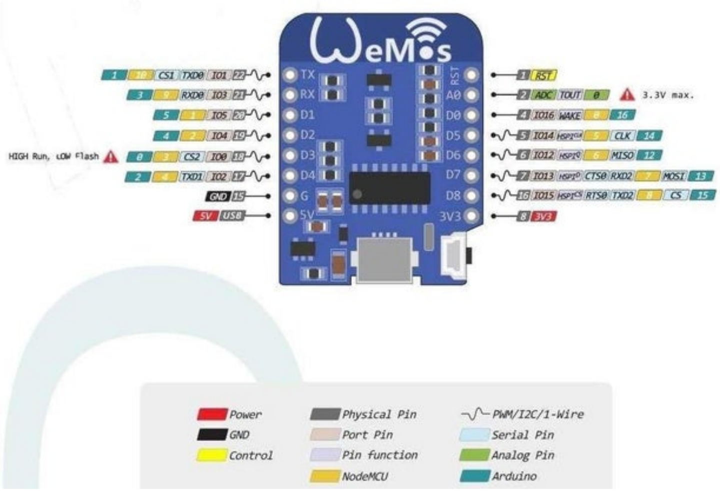
Phisical layers as you can see in the image are very simple, I add some additional logic layer.

First I create a library to manage a full set of informations of the inverter from the interface RS-485 available, than I create a series of thread (simulated) with specified delay to get data and store they in an SD in JSON format, than I create a full set of REST api to retrieve this set of information, a WebSocket server for realtime data, and a responsive web app to show all this data finally a configurable notification system via mail.



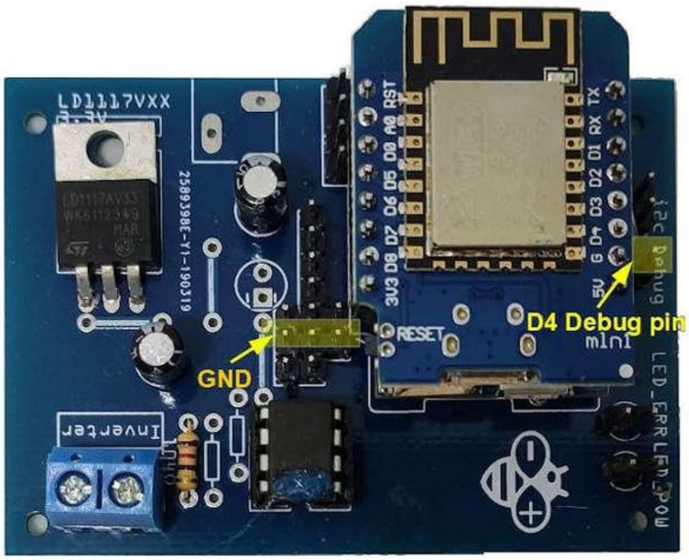
Step 5: Microcontroller

My selected microcontroller is an WeMos D1 mini, I choice this esp8266 device because It's very low cost and have sufficient specs to do all features I have in my mind. Here a mini guide on how to configure your IDE "WeMos D1 mini (esp8266), pinout, specs and IDE configuration".



Step 6: Debug the Device

I think that an interesting thing is that It has more Hardware Serial, so you can use Serial for communication with Inverter and Serial1 D4 (only Transmission) to debug. You can check how to connect debug USBtoTTL device on "WeMos D1 mini (esp8266), debug on secondary UART".



Step 7: Store Static Content and Data

We are going to put WebServer data in SPIFFS, the size needed is less than 2Mb. SPIFFS is explained in this article "WeMos D1 mini (esp8266), integrated SPIFFS Filesystem".

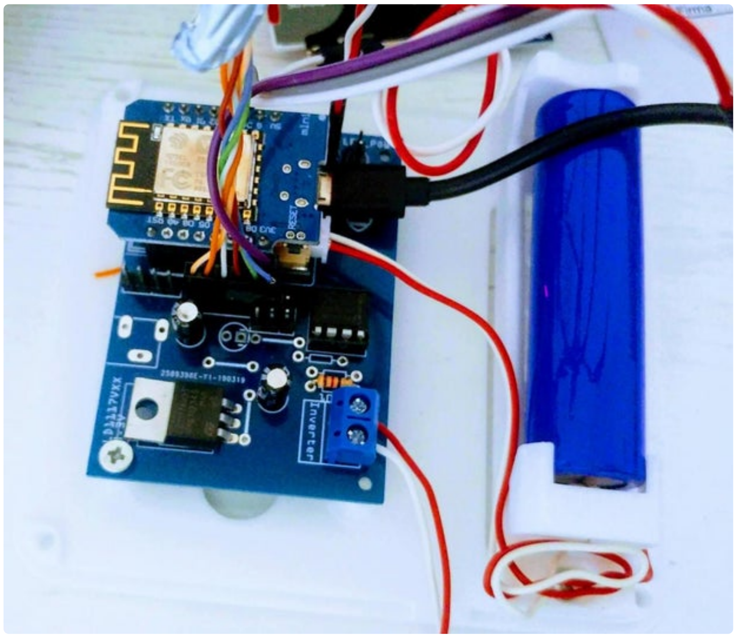
To update WebServer pages I use an integrated FTP server "FTP server on esp8266 and esp32". To store logging data we must add an SD card, It's not sure use SPIFFS (exist a 16Mb version of esp8266) because have a write cycle limitation. You can connect directly via an SD adapter, but I prefer a module to better fit in my case. You can find information on how to connect SD card in this article "How to use SD card with esp8266, esp32 and Arduino".

Step 8: Communication Via RS-485

Aurora ABB (ex PowerOne) communicate via RS-485 connection (more info how to do that "How to interface Arduino, esp8266 or esp32 to RS-485"), so the most important features is the communication protocol, and for first I create a complete library to interface on this interface via Arduino, esp8266 or esp32 device "ABB Aurora PV inverter library for Arduino, esp8266 and esp32".

Step 9: Simple Internal UPS

I use a 18650 rechargeable battery as UPS to grant server active when It's nigth and there aren't energy production, I use the schema from this article "Emergency power bank homemade".

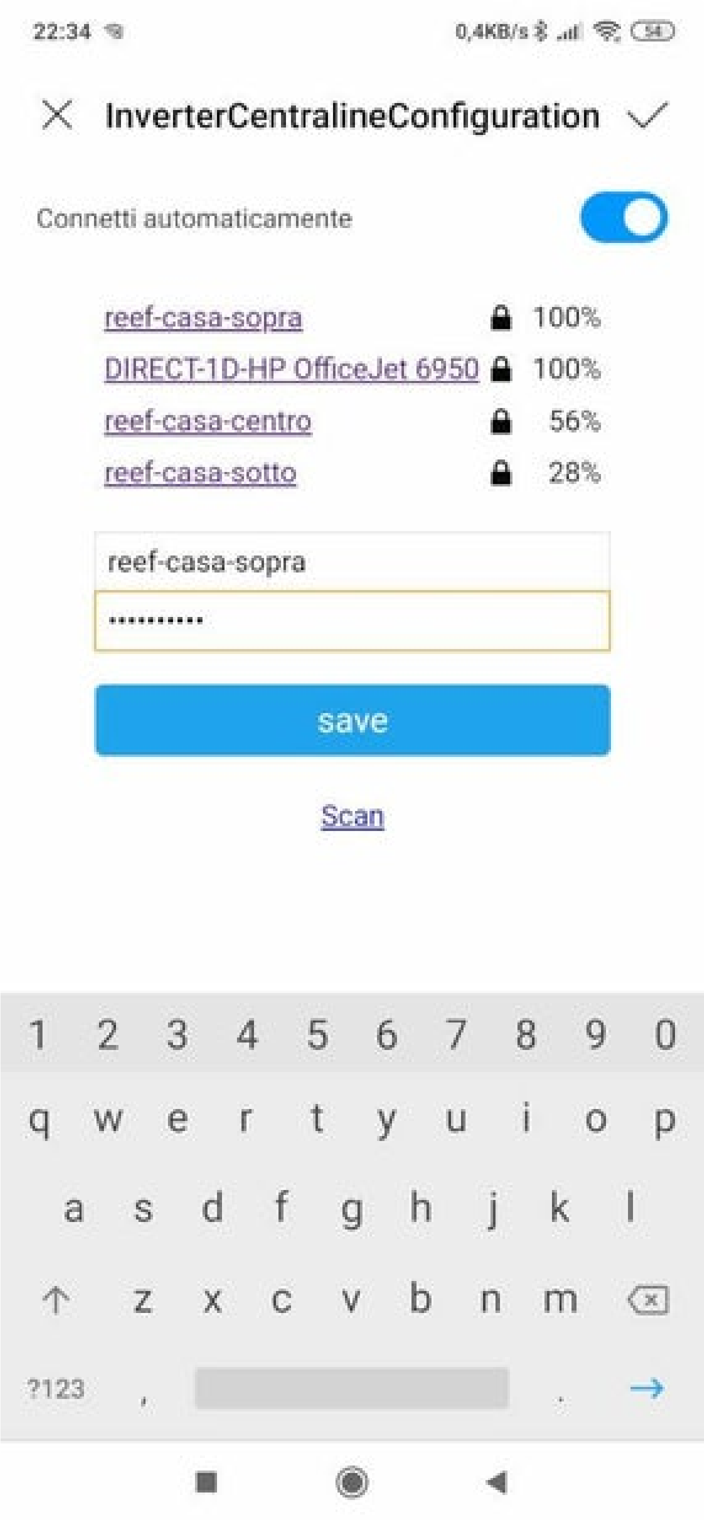


Step 10: Manage Date and DST

To logging data It's also important get current date and time, so I choice to try to get data from NPT server, if It isn't possible I get data from internal clock of inverter, I explain NTP with DST on this article "Network Time Protocol (NTP), Timezone and Daylight saving time (DST) with esp8266, esp32 or Arduino".

Step 11: Dynamic WiFi Configuration

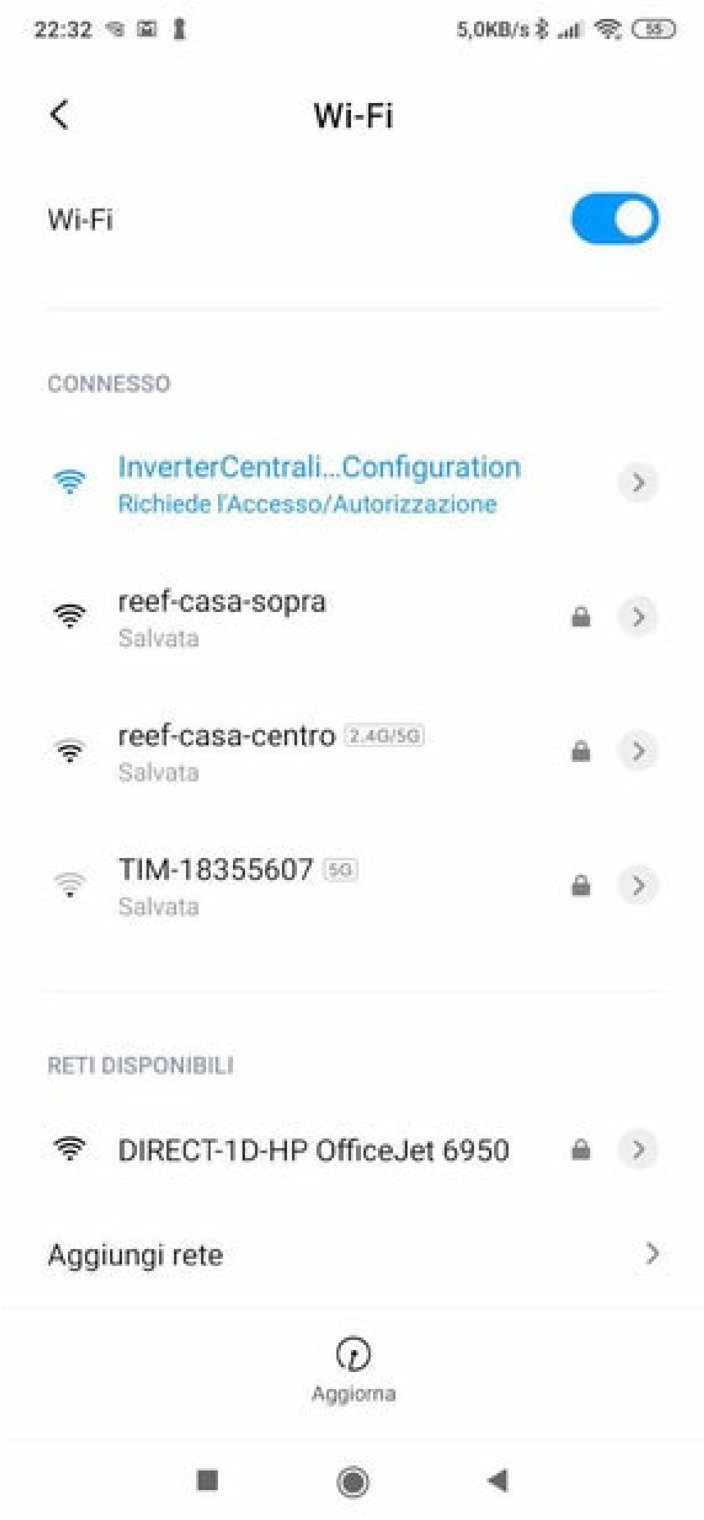
To connect device I use and fix WIFIManager thar start esp8266 as Access Point and give an interface to set connection parameter, for this argument you can refer to this article "How to manage dynamic WIFI configuration on esp8266 or esp32".



reef-casa-sopra	100%
DIRECT-1D-HP OfficeJet 6950	100%
reef-casa-centro	56%
reef-casa-sotto	28%

reef-casa-sopra

•••••••••••



InverterCentrali...Configuration
Richiede l'Accesso/Autorizzazione

reef-casa-sopra
Salvata

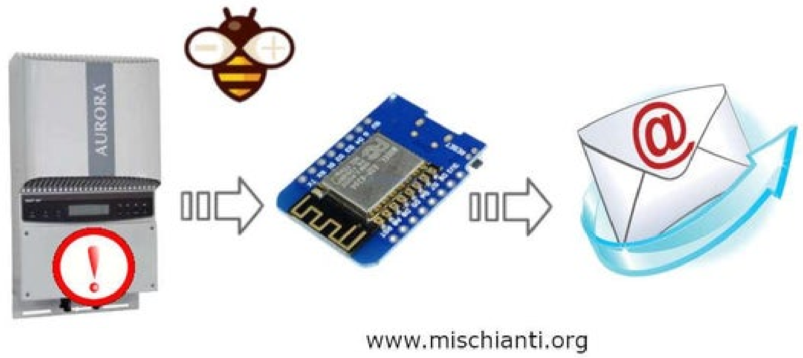
reef-casa-centro 2.4G/5G
Salvata

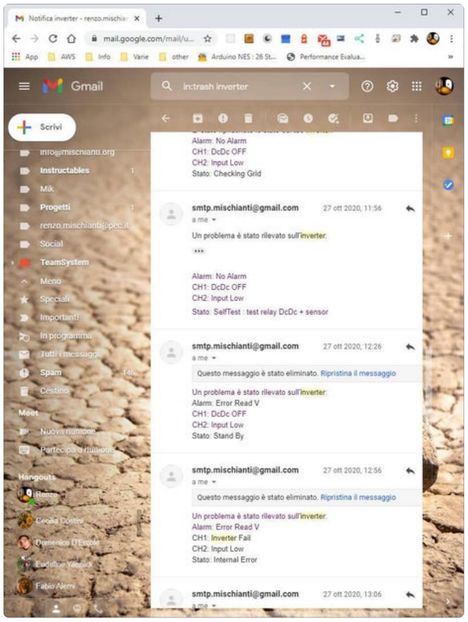
TIM-18355607 5G
Salvata

DIRECT-1D-HP OfficeJet 6950

Step 12: Notification Via EMail

I also create a library EMailSender to manage an SMTP server to send email when a problem raised in the inverter, I explain better this features (how to configure and where to get the library) on this article "ABB Aurora web inverter centraline (WIC): debug and notification".





Alarm: No Alarm
CH1: DcDc OFF
CH2: Input Low
Stato: Checking Grid

smtp.mischianti@gmail.com 27 ott 2020, 11:56
a me

Un problema è stato rilevato sull'inverter:

...

Alarm: No Alarm
CH1: DcDc OFF
CH2: Input Low
Stato: SelfTest : test relay DcDc + sensor

smtp.mischianti@gmail.com 27 ott 2020, 12:26
a me

Questo messaggio è stato eliminato. Ripristina il messaggio

Un problema è stato rilevato sull'inverter:
Alarm: Error Read V
CH1: DcDc OFF
CH2: Input Low
Stato: Stand By

smtp.mischianti@gmail.com 27 ott 2020, 12:56
a me

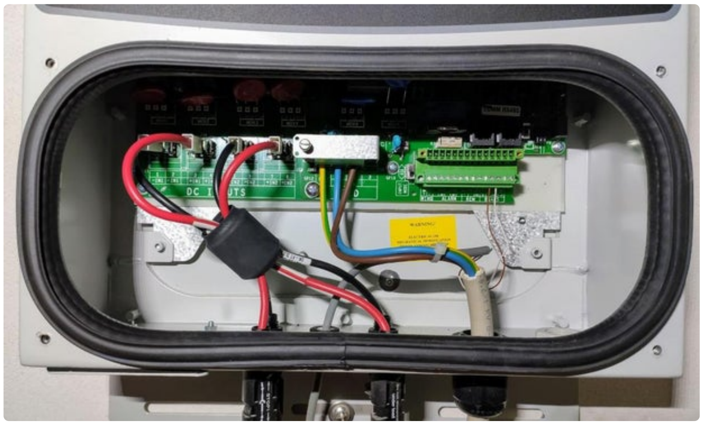
Questo messaggio è stato eliminato. Ripristina il messaggio

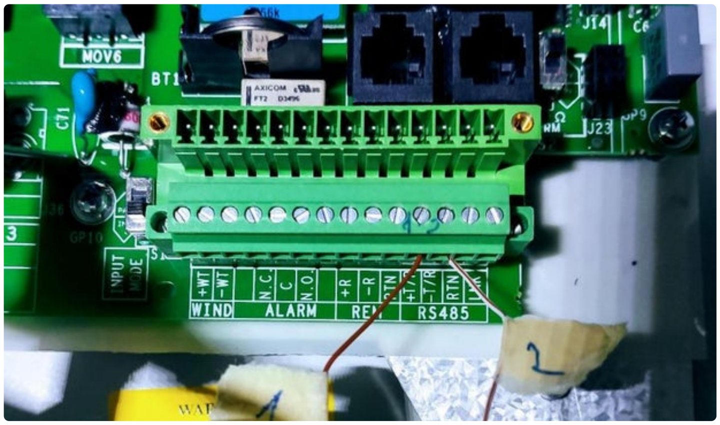
Un problema è stato rilevato sull'inverter:
Alarm: Error Read V
CH1: Inverter Fail
CH2: Input Low
Stato: Internal Error

smtp.mischianti@gmail.com 27 ott 2020, 13:06
a me

Step 13: Inverter Wiring

As I already say, i use the RS-485 protocol via MAX3485 IC to wiring the inverter, only 2 wire is needed, but you must open your inverter, and check the address, you can find the instruction on "ABB Aurora Web Inverter Monitor (WIM): wiring Arduino to RS-485".





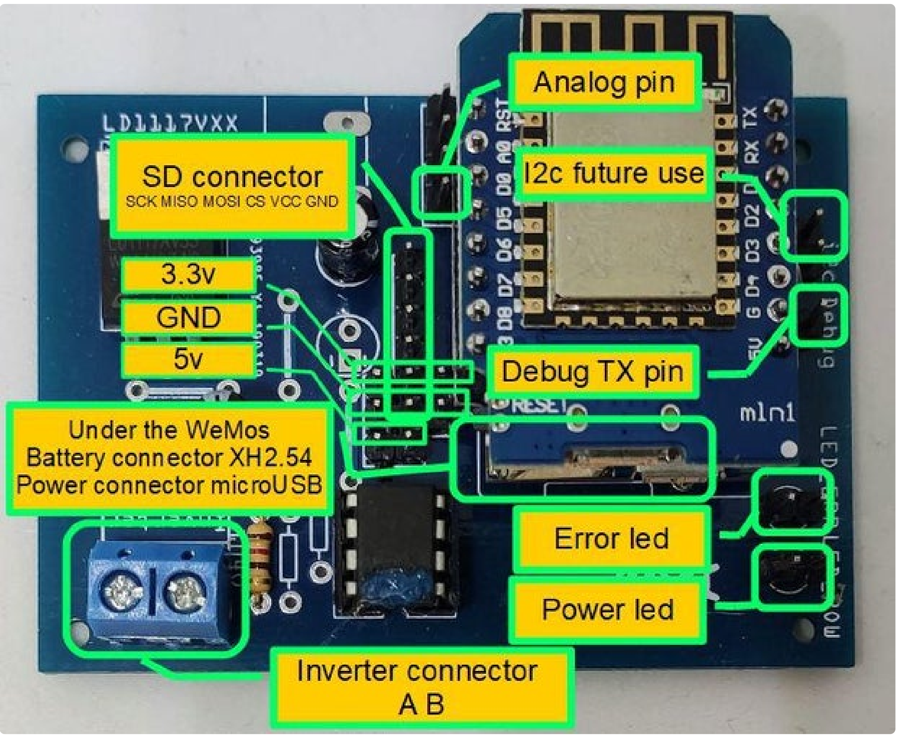
Step 14: PCB

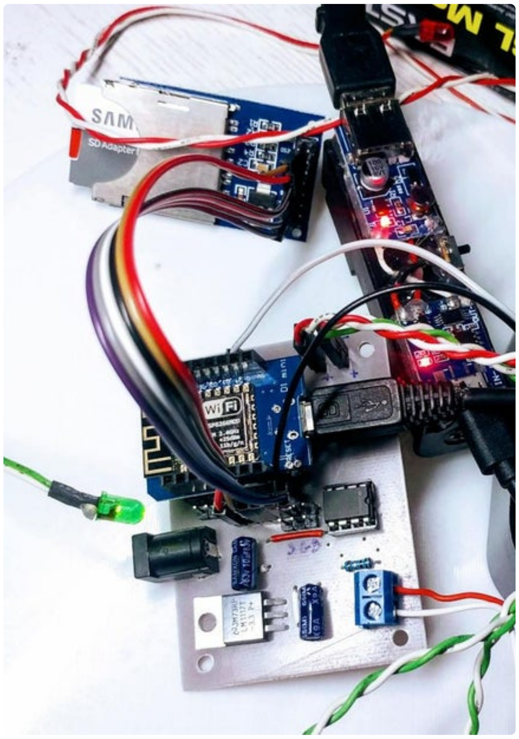
The PCB is not so complex, but I share It for free, (5$ 10 PCB) so all people can do this project with few list of component, I add the order instruction of the PCB here "ABB Aurora Web Inverter Monitor (WIM): wiring and PCB soldering ".

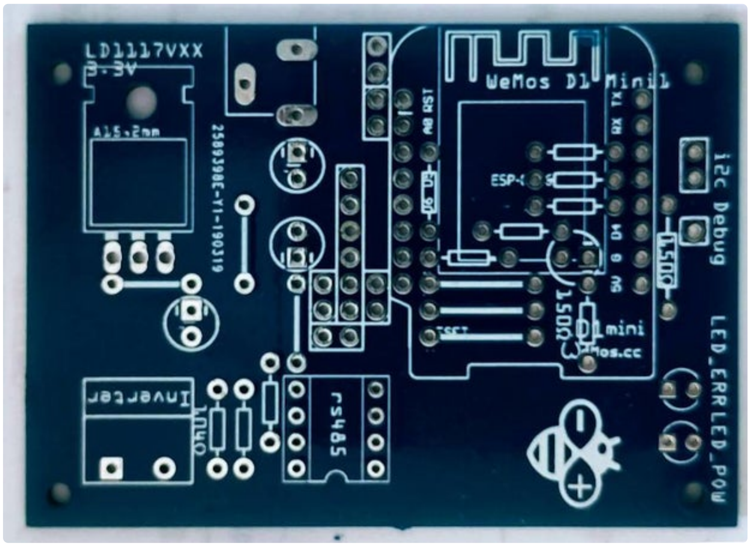
You can order without additional cost at this PcbWay link.

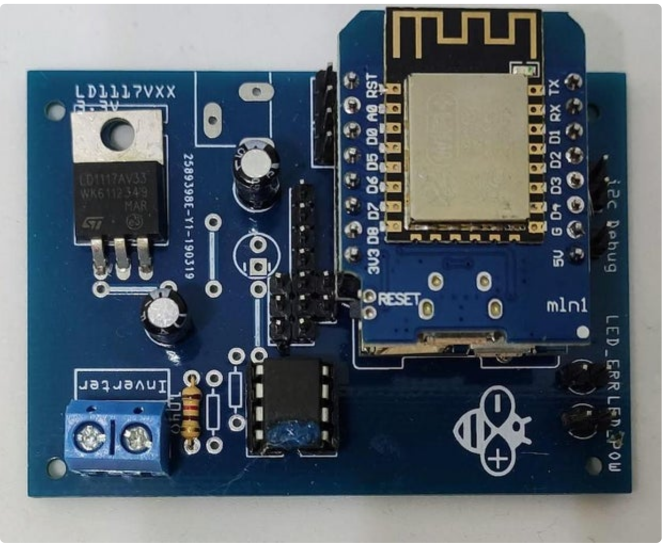
PCB

- 2 Electrolytic Capacitor: capacitance 10µF
- 1 Generic male header – 1 pins: pin spacing 0.1in (2.54mm); form (male)
- 6 Generic male header – 2 pins: pin spacing 0.1in (2.54mm); form (male)
- 1 Camdenboss CTB0158-2: pin spacing 0.2in (5.08mm); hole size 2.7mm; variant 90° 2 connector;
- 1 Generic male header – 6 pins: pin spacing 0.1in (2.54mm); form (male)1
- 104Ω Resistor: pin spacing 400 mil; resistenza 104Ω; tolerance ±5%; bands 4; package THT2
- 150Ω Resistor: pin spacing 400 mil; resistenza 150Ω; tolerance ±5%; bands 4; package THT1
- MAX485
- LD1117V33: voltage 3.3V; package 78xxl; chip LD1117VXX
- 1 WeMos D1 Battery shield
- 1 WeMos D1 Mini: cpu ESP-8266EX;









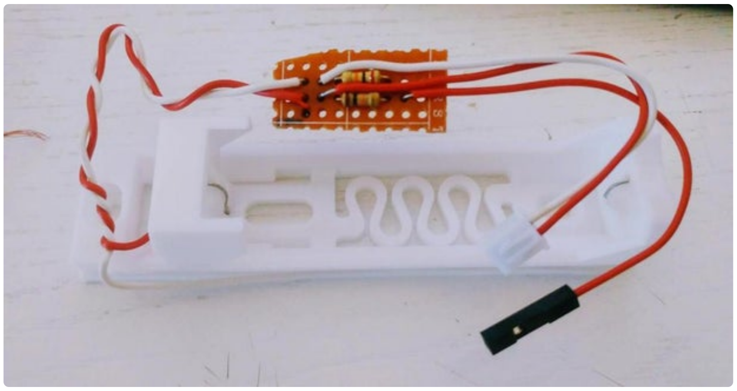
https://www.youtube.com/watch?v=wCU95Tzlq34

Step 15: Battery Adapter and SD Card Module

You need also to do a simple voltage divider for the battery check.

Battery adapter(check the video)

SD card module Detailed BOM, assembly video and other information in the article "ABB Aurora Web Inverter Monitor (WIM): wiring and PCB soldering".

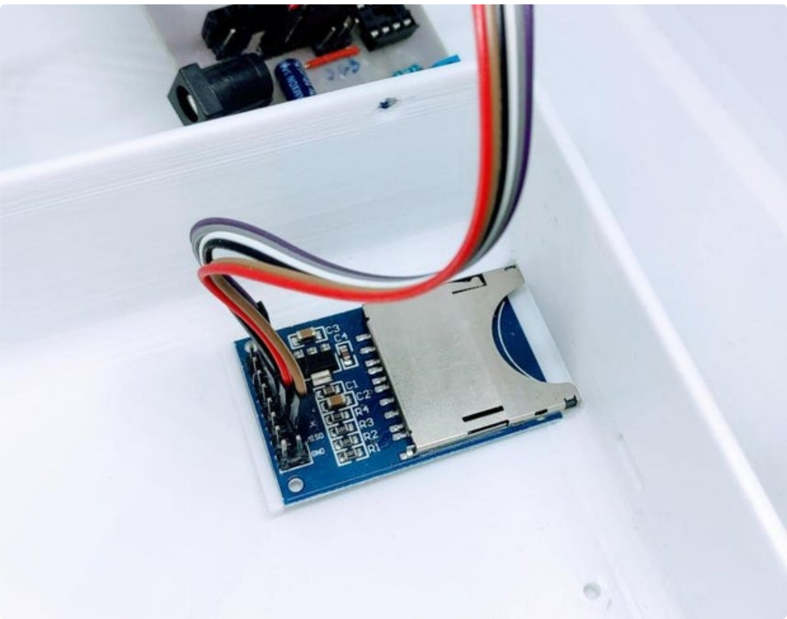


https://youtu.be/kTFTxbHQJ6Q

Step 16: Case

I must do a case to prevent dust and other problem, so I creted one 3d printed, that you can download from here "ABB

Aurora web inverter monitor (WIM): 3D printed case to complete project"



Step 17: Upload Sketch and Front End

You can get the source code from GitHub repository.

There are a long list of library that you must add to your arduino ide, I write a simple article to help you on that operation "ABB Aurora Web Inverter Monitor (WIM): upload the sketch and front end ".

- ArduinoJson
- ArduinoThread aurora_communication_protocol
- DNSServer
- EMailSender

- ESP8266mDNS
- ESP8266SdFat
- ESP8266WebServer
- ESP8266WiFi
- Hash
- NTPClient
- SD
- SDFS
- SPI
- TimeLib
- Timezone
- WebSockets
- WiFiManager
- Wire

Step 18: Web Interface

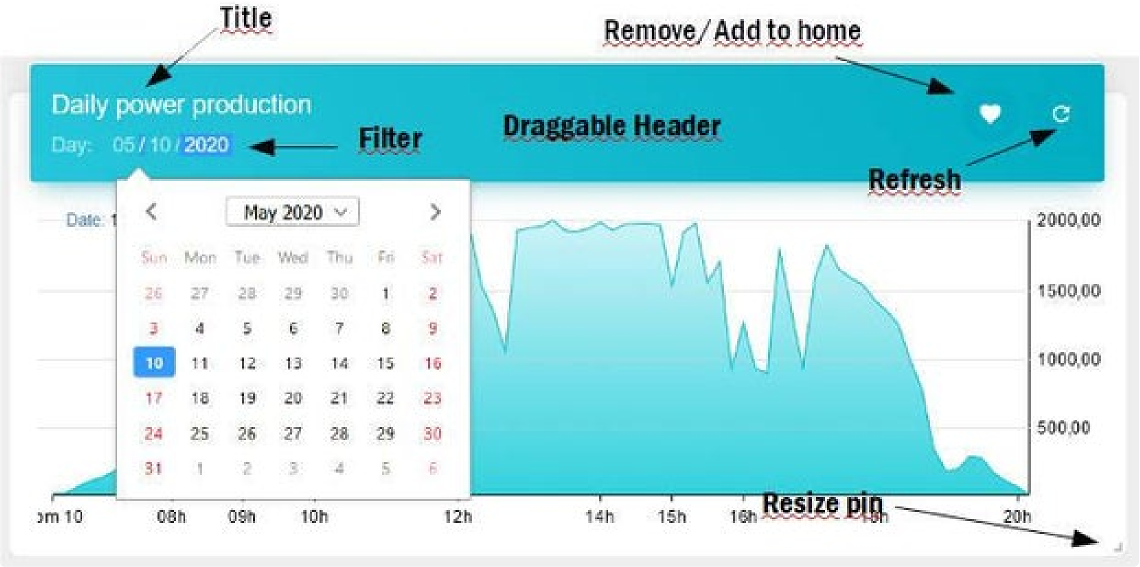
The interface is subdivided in various section explained in detail here "ABB Aurora Web Inverter Monitor (WIM): WebSocket and Web Server"

- Homepage: where you can add all single widget to create a personalized view
- Daily chart: with
    - chart of power
    - chart of current
    - chart of voltage
    - total daily production
    - realtime production

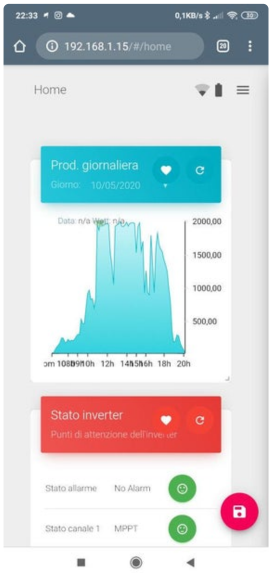
- Monthly chart: with
    - chart of monthly production, for the month you can view every day the total of production and the peak of the production
    - Total lifetime prduction
    - Total yearly production
    - Total monthly production
    - Total weekly production

- Information and state: with
    - chart of the battery
    - voltage Inverter info
    - Inverter state

- Configuration


Daily power production

Day: 05/10/2020



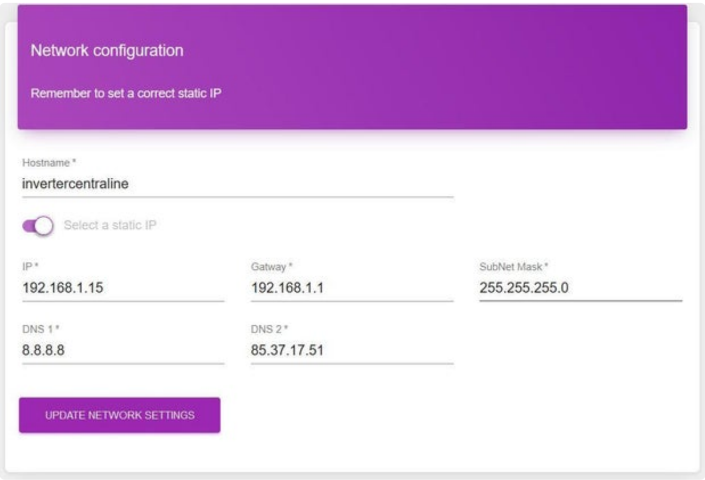
Prod. giornaliera

Giorno: 10/05/2020

Stato inverter

Punti di attenzione dell'inverter

Stato allarme	No Alarm
Stato canale 1	MPPT


Network configuration

Remember to set a correct static IP

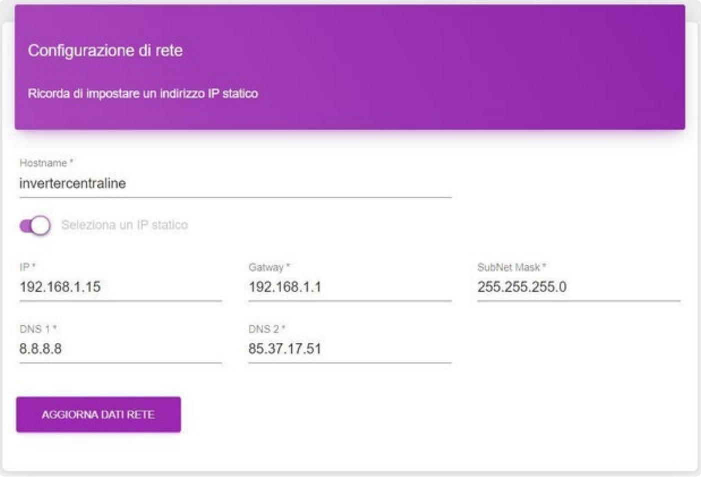
Hostname *
invertercentraline

Select a static IP

IP *	Gatway *	SubNet Mask *
192.168.1.15	192.168.1.1	255.255.255.0

DNS 1 *	DNS 2 *
8.8.8.8	85.37.17.51

UPDATE NETWORK SETTINGS


Configurazione di rete

Ricorda di impostare un indirizzo IP statico

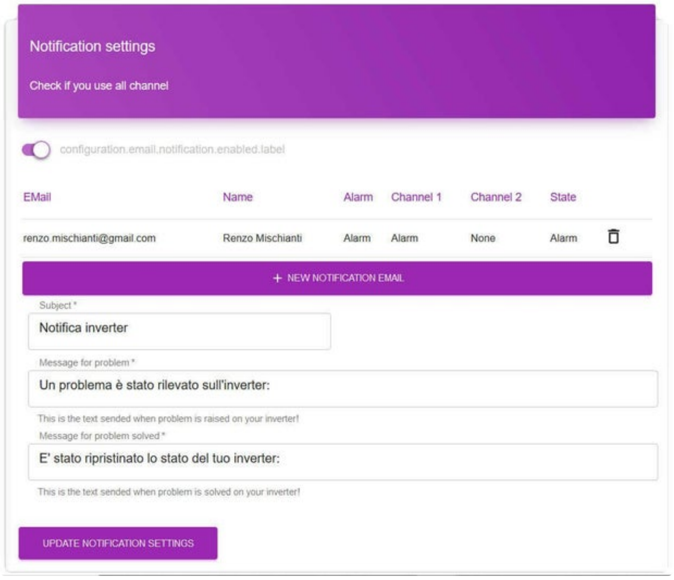
Hostname *
invertercentraline

Seleziona un IP statico

IP *	Gatway *	SubNet Mask *
192.168.1.15	192.168.1.1	255.255.255.0

DNS 1 *	DNS 2 *
8.8.8.8	85.37.17.51

AGGIORNA DATI RETE


Notification settings

Check if you use all channel

configuration.email.notification.enabled.label

EMail	Name	Alarm	Channel 1	Channel 2	State
renzo.mischianti@gmail.com	Renzo Mischianti	Alarm	Alarm	None	Alarm

+ NEW NOTIFICATION EMAIL

Subject *
Notifica inverter

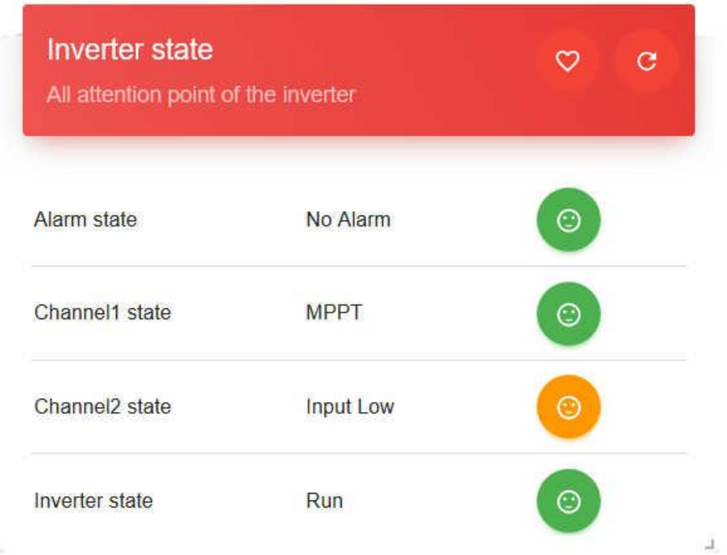
Message for problem *
Un problema è stato rilevato sull'inverter:

This is the text sended when problem is raised on your inverter!

Message for problem solved *
E' stato ripristinato lo stato del tuo inverter:

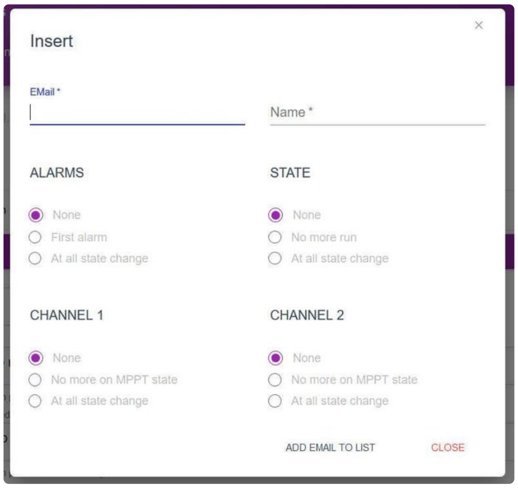
This is the text sended when problem is solved on your inverter!

UPDATE NOTIFICATION SETTINGS


Inverter state

All attention point of the inverter

Alarm state	No Alarm
Channel1 state	MPPT
Channel2 state	Input Low
Inverter state	Run



Insert

EMail *

Name *

ALARMS

● None
○ First alarm
○ At all state change

STATE

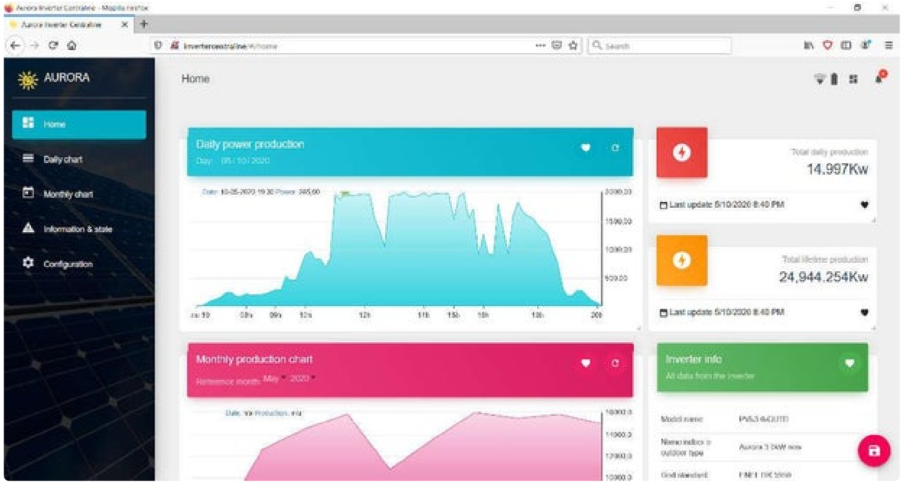
● None
○ No more run
○ At all state change

CHANNEL 1

● None
○ No more on MPPT state
○ At all state change

CHANNEL 2

● None
○ No more on MPPT state
○ At all state change



Step 19: Thanks

All detailed information

1. ABB Aurora Web Inverter Monitor (WIM): project introduction
2. ABB Aurora Web Inverter Monitor (WIM): wiring Arduino to RS-485
3. ABB Aurora Web Inverter Monitor (WIM): storage devices
4. ABB Aurora Web Inverter Monitor (WIM): debug and notification
5. ABB Aurora Web Inverter Monitor (WIM): set time and UPS
6. ABB Aurora Web Inverter Monitor (WIM): WIFI configuration and REST Server
7. ABB Aurora Web Inverter Monitor (WIM): WebSocket and Web Server
8. ABB Aurora Web Inverter Monitor (WIM): Wiring and PCB soldering
9. ABB Aurora Web Inverter Monitor (WIM): upload the sketch and front end
10. ABB Aurora web inverter Monitor (WIM): 3D printed case to complete project
11. ABB Aurora web inverter monitor (WIM): repair E013 error

- GitHub repository with all code BE and FE transpiled