Soggy Paws' Notes on the Electrodacus BMS

v.002

Change Log

v.002	03/07/2021	Added section on Shunt Zero Calibration, and a few more resource links

1 The SBMS0 System Overview

<Source: Most of Section 1 comes from this source: https://github.com/bdlow/sbms0 . I don't think this guy ever updated it after he used the BMS (if he did in fact use it), this is just research, and may not actually be factual. My guess he was taking notes while trying to figure out which BMS to buy, and eventually bought something else.>

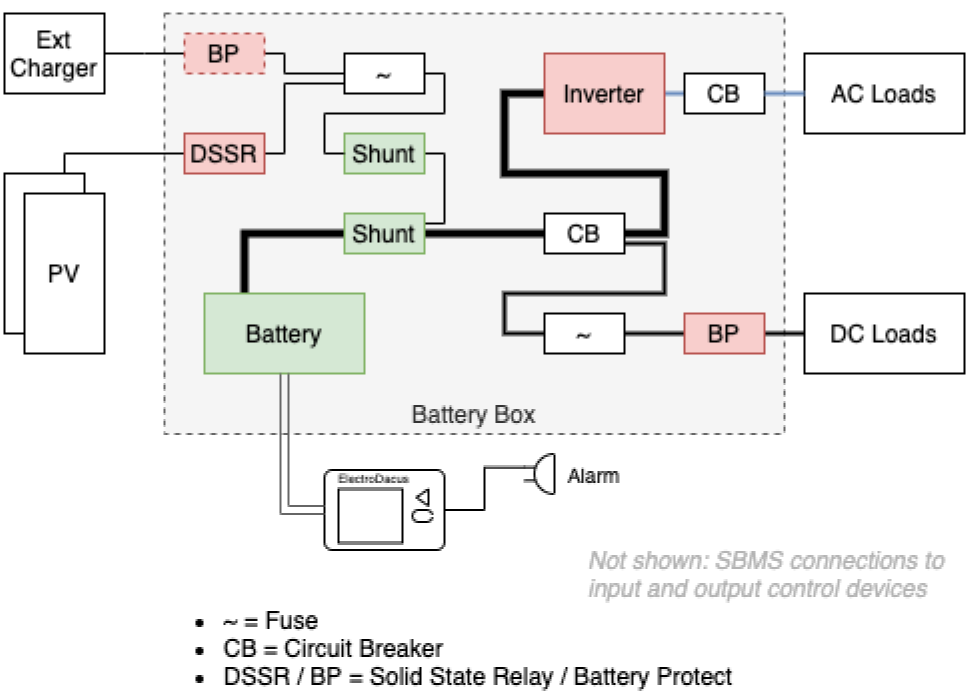
The SBMS0 system is a little different than some all-in-one Battery Management Systems (BMS) that incorporate the control and power connect/disconnect within the one unit. Separating out the control and power functions provides greater modularity and flexibility in design and installation, and greater durability (the components most likely to fail - the power switching - are more readily repaired or replaced when they're separate modules). The SBMS0 is the control unit, the Diode + Solid State Relay (DSSR20) is the nominal input power switching component (others such as a Victron BatteryProtect can be used).

The following diagram shows a basic PV power system, consisting of:

Charging sources (PV and external charger, e.g. DC-DC charger)

"Battery box" with various fusing, control and measurement functions

Loads (AC and DC)



The items in green are measured by the SBMS0 (battery cell voltages, load and charge currents), and those in red are controlled by the SBMS0. The remaining components within the battery box are protective devices.

1.1 Why It is Unique

The SBMS0 is a novel approach to managing solar-powered energy storage, produced by ElectroDacus as an open-source hardware project (as of mid 2020 some hardware details such as PCB layout and the software source code are not yet published). Dacian Todea, the project's lead and primary (sole?)

contributor, has been developing various iterations of a solar battery management system since at least 2011. Dacian's not in the business of producing SBMS systems at commercial scale.

The SBMS0 is the fourth (?) major design iteration and departs from earlier designs, and most commercial BMS, in separating out the controller from the power switching. The primary novelty for the SBMS0 is in not using any form of DC-DC conversion between the solar photovoltaic (PV) panel(s) and the battery - the principle being that a PV input suitably matched to suitable battery chemistry (e.g. $LiFePO_4$) does not actually require any complex DC-DC conversion. Conventional wisdom is that some form of DC-DC conversion, such as a Maximum Power Point Tracking (MPPT) solar controller, is required to maximize power transfer from the PV to battery and/or safely charge the battery (e.g. lead-acid batteries have very particular charging requirements to maximize service life).

However the SBMS0 approach recognizes that, whilst valuable and even necessary for some sub-optimal combinations of PV and battery, for an appropriately designed system with modern battery chemistry a DC-DC converter (MPPT controller) has marginal benefit at not-insignificant cost over a simpler direct connection. These costs are mainly financial and reliability (both short and long term). If the funds that would have been spent on an MPPT controller are put towards over-provisioning PV panels the result can be an equally or higher capacity system that is cheaper and simpler to install and operate and with a much longer service life. Even when over-provisioning is not practical (e.g. on a space-constrained installation such as a van), the marginal capacity "loss" from not using an MPPT is not generally a problem and still provides simplicity and durability benefits.

Further, the SBMS0 system design can direct any excess energy from the over-provisioned PV input to thermal storage (heating).

1.2 Controller Feature Summary

SBMS0 system features:

1. durable: no complex power electronics; the DSSR20 is a simple on/off cutoff with no power capacitors
2. input capacity: up to 600A @ 49V; practical PV limit: 60 x 72-cell panels ("30S2P"), approx. 24kW
3. battery capacity: 8-32 V; $LiFePO_4$: 3-8S
4. long-term data logging: up to 1 year at 2 minute granularity
5. very efficient: controller power consumption is less than 1W, switch power loss is less than 1.5W per module; passively cooled
6. cell balancing up to 200mA
7. no direct load disconnect - requires separate component(s) and/or devices with appropriate remote on/off control

1.3 Controlling Power In

The SBMS0 controller unit, in conjunction with one or more distributed PV disconnect modules, such as the DSSR20, manages the solar input to ensure that the PV panels are disconnected when the battery is fully charged. It calculates a battery State of Charge (SoC) by measuring each individual cell voltage as well as current in and out of the battery via an external shunt. If an optional second shunt is used, it can also separately measure PV current input - this is useful mainly for measuring system performance and not required for battery management per se.

1.4 Controlling Power Out

On the discharge side of things, the SBMS0 can control external low voltage disconnect components such as an inverter remote control or load relay/SSR (e.g. Victron BatteryProtect). **The SBMS0 cannot, alone, disconnect loads on a low-SoC state.**

1.5 Input (PV) Disconnect

The power generated by a PV panel has an interesting characteristic in that current approaches zero as the voltage reaches maximum. If the panel were perfectly matched to the battery voltage and chemistry, it could simply be permanently directly connected to the battery as the panel wouldn't be able to go beyond the battery's safe operating voltage. In practice variations arising from temperature and irradiance mean that such a setup would not be capable of fully charging a battery except in exceptional "ideal" conditions. Practically, the PV panel should be such that it can deliver a suitable charge current even under non-ideal conditions, which means that in better conditions it would be capable of over-charging the battery.

Thus, in a practical direct-connection setup, some means of disconnecting the charging source (PV input or other) is necessary to prevent damaging overcharging. The SBMS0 is designed to work with one or more DSSR20 which are an inexpensive and reliable method of disconnecting PV or other inputs up to 20A (two large 72-cell panels). The SBMS0 can handle up to 30 DSSR20 modules for a total of 600A (29kW at the maximum input voltage of 49V; for PV inputs the limit is 60 x 72-cell ~400W panels with two panels per DSSR20, approx. 24kW).

Alternatively to the DSSR20, any charging source with a suitable remote on/off can be used. For example, a DC-DC vehicle charger.

1.6 Input (PV) Diversion

The DSSR20 is available in two versions: with and without diversion. The latter is slightly cheaper as it omits some components. The diverting version of the DSSR0 can switch two loads: the primary (battery) as above, and a second "diverted" load. The intention is that when excess PV power is available that excess power can be connected to the other load such as thermal storage.

The companion DMPPT450 project extends the diversion concept for larger-scale heating; the DMPPT450 sits in front of the SBMS0 in place of the DSSR20 switches and can selectively connect groups of panels and the battery and thermal sinks to optimize the energy transfer between them.

1.7 Load Disconnect

On the load side of the system it's critical to prevent battery over-discharge. For some battery chemistries such as LiFePO4 an over-discharge event will lead to permanent loss of capacity. When the battery reaches a sufficiently low SoC it's important to disconnect all loads before damage occurs. For the SBMS0 this is accomplished via signalling other components to disconnect - for example, the remote on/off function of an inverter or load connection relay.

Aside on inverter disconnects: highly-capacitive loads such as inverters are challenging to switch on safely, the large inrush currents that occur when they are first powered on can damage even the sturdiest of relays, SSRs have no hope. Few load switching or battery disconnect systems are rated to switch inverters. It's often recommended that inverters and their ilk be connected permanently to the battery (via a fuse or circuit breaker only), and a low-power remote on/off control used to switch them on and off.

1.8 Connections

The SBMS0 has multiple sets of connections:

1. Cell Monitoring and Balancing (grey 12-way IDC ribbon cable on the rear of the module)
2. Input/Output (IO) connector for measuring current flow and temperature, and controlling load disconnects (10 pin wire "quick connect" on the right hand side of the module)
3. WiFi/USB - normally populated if you order the WiFi/USB version (2x8 pin header)
4. Isolated Data - an unpopulated header on the WiFi/USB board that presents isolated I2C, serial data and extra EXTIO5,6

The SBMS0 is provided with a 2x5 pin IDC box header that may be soldered to the Isolated Data connection; it's not normally installed to keep the physical profile of the module to a minimum

1.9 Cell Monitoring

Cell monitoring is performed by a dedicated 3-to-8 Cell Li-ion Battery Pack Monitor IC, ISL94203. The ISL94203 also handles cell balancing (see below).

The provided cell monitoring 12-way ribbon cable is 28AWG (0.07mm^2) and does not need to be separately fused; any significant over-current will fuse the wire without pyrotechnics. The ISL94203 can detect an open connection to a cell (TODO: unclear if this function is used by the SBMS0: evidently the ISL94203 configuration registers are displayed by the SBMS0 diagnostic: check register 4A).

The ISL94203 only reads its configuration on power-on- thus any parameter changes made on the SBMS0 will only take effect when the SBMS0 is power cycled by disconnecting and reconnecting the cell monitoring cable.

The 12-way cable also supplies power to the SBMS0, from the battery being monitored. The power supply wires 1 (GND) and 11+12 (Positive) should be kept electrically separated from their neighboring sense wires up to the battery so as to ensure the SBMS0 current draw does not affect the measured cell voltages. i.e. don't "piggyback" wires 1-2 or 10-12.

1.10 How Well the SBMS0 Fits the Typical Cruising Boat Scenario

The SBMS0 was designed for off-grid solar conditions. The developer lives in a house in the woods in Saskatchewan Canada and has presumably never been on a boat in his life. So there is little understanding of the need for redundancy—ie how bad it would be if the BMS disconnected the battery in the middle of a storm at night, for example. He also dislikes MPPT Solar Charge Controllers (thinks they are a waste of money). And he uses a good part of his excess solar to try to warm his house. So a good part of the design reflects this frame of reference (naturally).

If you are just starting out with solar on your boat, and fit your boat with solar panels that match Dacian's vision, and use his DSSR's to cut them off, have fairly well matched cells (and/or a fairly small bank) and are happy with one 12 or 24v battery bank, his BMS is an inexpensive solution.

People with Victron solar controllers and inverters seem to be going with the Rec BMS, which has some Victron integration. But the Rec BMS is quite a bit more expensive than the Electrodacus.

2 Shunts for Measuring Current In and Current Out of Battery

The SBMS0 is designed to use conventional current shunts for measuring current to/from the battery, and optionally from the charging source. If the charging (PV) shunt is installed, the SBMS0 is then able to distinguish charge and load currents separately which can then provide an accurate view of PV performance independently of loads. If the PV shunt is omitted, the SBMS0 can still provide battery management and State of Charge (SoC) measurements.

Shunts must be selected to be able to carry the expected current however larger rated shunts will result a more coarse measurement resolution by the SBMS0, and thus a less accurate SoC calculation. Most shunts must be rated to at least 150% of the expected load, 200% for elevated temperatures - refer to the shunt manufacturer for detailed derating specifications.

2.1 Are Shunts Required?

Question: Is it even required, technically, to use shunts at all? As I understand it, high and low cutoffs are based on voltage - the shunts just provide useful monitoring. But in theory, if one didn't care about that, would it be possible to use an SBMS setup without any shunts?

Answer: The battery will still be protected even without shunts but there will be no cell balancing as that only happens when SBMS0 sees a minimum of 300mA of battery charge current.

Also the SOC cannot be calculated thus you do not know the state of charge of your battery. SOC is super important info, and so you should have at least the battery shunt to get this and cell balancing.

You can get a cheap current shunt for just 10 or $15 and if you also use the PV shunt you get extra info like PV energy production and Load consumption else with just one shunt you only have the battery current so just the SOC calculation.

2.2 What Shunts to Buy

<google groups post by Dacian June 23, 2020>

I get quite a few question related to current shunts so I decided to make this post.

SBMS0 supports any shunt between 0.0400mOhm and 9.9999mOhm that means all shunts in the Riedon RS range https://riedon.com/media/pdf/RS.pdf except for the smallest 5A one but including 1200A.

As I order parts for SBMS0 production I also ordered a few of the Reidon shunts one from each category the RSA smallest ones up to 150A so I got the 100A 100mV one then one from the midd size RSB and got the 500A 100mV one and finally the huge RSC 1000A 100mA.

Interestingly the 500A 100mV was the only one coming with a certificate of calibration showing the exact current shunt value of 0.200045mOhm so that will round out to exact 0.2000mOhm in the SBMS settings.

I got my current shunts from Digikey that should be available in almost any country https://www.digikey.ca/

But Riedon also has his own website and I think you may be able to buy directly from them https://shunts.com

Murata also has a similar line of current shunts and I have the small 10A 100mV you can see in one of the photos below.

I attempted to measure the current shunt resistance but I do not have precision enough power supply and multimeter to get to the level of precision required to check their calibration (not even close). I have used a constant current power supply set at 3.0065A measured with Wavetek model 52 bench multimeter the last digit was not stable on that power supply so it can be in the 3.0060 to 3.0070 range the 3.0065 was sort of the average and where it was most of the time.

In any case the table below contains the results

On the left side you have the spec and on the right side the voltage reading in mV at the fixed 3.0065A current then calculated shunt resistance based on that result and then calculated error.

Even with my limited measurement accuracy and resolution all the Riedon and Murata shunts where in spec below the 0.25% error while the two shunts from China had an order of magnitude larger tolerance.

The 100A 75mV shunt was looking like calibrated as it had a cut but it is even more out of spec than the 200A 75mV one that had no calibration done and it makes sense that had no calibration as the value of resistance was already higher than spec so they can not calibrate by removing material as that will make things worse.

In any case even the low cost China shunt are usable if you are on a budget since you can easily calibrate the error out using the SBMS0 settings. So you just compare the reading of the SBMS0 with a known good multimeter and adjust the resistance value until your multimeter measurement and SBMS0 measurement match.

But there are other advantages to the Riedon and Murata shunts as they already have an insulated base and they are more compact. And a shunt will always have value even in 10 or 20 years shunts will likely not change as they are just a power resistor and so you can reuse tose for decades.

While you can use the 1000 or even 1200A current shunt, the SBMS0 will limit the max reading to 750A

Also max continues for all shunts is 66% of the rating as you can see in spec so if you have say a 600A continues load you will need the 1000A shunt.

A	mV	R spec	mV reading	R measured	% error
Riedon			3.0065		
1000	100	0.1000	0.300	0.099783802	0.216
500	100	0.2000	0.600	0.199567604	0.216
100	100	1.0000	3.003	0.998835856	0.116
Murata					
10	100	10.0000	30.010	9.981706303	0.183
China					
100	75	0.7500	2.185	0.72675869	3.099
200	75	0.3750	1.142	0.379843672	-1.292

Not sure how clear this picture is (below), but the bottom screw heads are deep and will not touch the base you install this on and there is a clear resin on top of them.

SBMS0 for size reference vs Reidon Shunts

On the right side is the 10A 100mV Murata very similar to the RSA series and below the 200A and 100A 75mV inexpensive shunts.

The bottom of the 100A 100mV Riedon and the 10A 100mV Murata looks like they are not manufactured by the same company fairly significant differences.

Material seems to also be different but form factor is very similar.

2.3 50mV vs 75mV vs 100mV Shunts

SBMS0 has an input range of 90mV that mean for example with a 100A 100mV shunt SBMS can measure up to 90A but since it uses the entire range measurement resolution will be very good.

An 100A 50mV shunt will have half the resolution in the same measurement range but also less heat dissipated on the shunt.

So say at 50A you will have 25mV drop on the 50mV shunt and 50mV drop on the 100mV shunt so 2.5W as heat on the 100mV shunt and half that 1.25W on the 50mV shunt.

So if you care most about accuracy you will go with the 100mV shunt and if you care more about heat (maybe cramped space and very high currents) then you may want to go with the 50mV shunt.

Of course a 200A 100mV shunt will have the same resistance as a 100A 50mV shunt but the 200A 100mV will be larger to deal with the extra heat that needs to be generated.

The 75mV where somewhere in the middle and is a very common value for those low cost shunts from China.

A 200A 100mV current shunt will have the same resistance (0.5000mOhm) as a 100A 50mV shunt so they will have the same resolution.

The 200A 100mV will be larger so that it can handle the increase amount of heat that needs to be dissipated at the higher current.

The SBMS0 ADC resolution is 16bit that means plus minus 15bit that will be 32768

So for example with the 200A 100mV shunt the SBMS0 can read only up to 90mV thus 180A max and for that there are 32768 steps.

180A / 32768 = 5.5mA will be the smallest step so resolution in this example will be 5.5mA that is the smallest change in current that can be read.

Bottom Line: Any shunt will work, all you need to know is the shunt resistance to set up in the SBMS menu. Usually the current rating and associated voltage drop is written on the shunt and from those you can calculate the shunt resistance.

2.4 Balancing Size of Shunts vs Measurement Accuracy

Question: I purchased a shunt recently that I thought I would share with the forum. I was having a hard time finding a shunt that took 5/16 or 3/8 battery cable lugs that was less than 200amps. My use case is

80 amps max charge/discharge. All the shunts rated less than 200amps seem to take only 1/4 inch lugs, which is awkward for large battery cables.

My typical loads are 0.7 amps to 3.5 amps, with a once a day, short duration 80 amp load and charging current typically about 70amps. I was unsure how sensitive the SBMS was was in detecting the shunt voltage drop, so was trying to keep as low as possible shunt value to capture the small loads.

Answer: There is not a problem using a 200A current shunt for max 80A charge discharge rate. My own system has a 200A 75mV current shunt and my max discharge rate is below 80A

There will not be much of a difference between a 150A and a 200A shunt in terms of measurement resolution.

That is also my typical Load with around 4 to 5A during the day and around 1A at night and the measurement resolution is more than sufficient with the 200A 75mV shunt I have.

Question: On the load side, I have "small" loads like lights, the radios (~20 amps), and stuff that normally will never get above 50 amps TOPS. I also have a 2kw peak inverter for AC loads.

Is there any way to hook up my SBMS0 with 2 separate shunts, one for the small loads, and a bigger one for the inverter?

Answer: The SBMS0 has only inputs for two current shunts. One will measure the battery current and the other will measure the PV current. Based on these two, the Load current is also calculated. I see no reason why you will want to see the inverter current separate from the DC loads.

The accuracy will be plenty good with a 200A shunt.

If you want best accuracy get a 200A 100mV shunt that will be able to display current in the range of plus minus 180A and with the 16-bit ADC resolution on the SBMS0 you have 15bit (32768) resolution in one direction.

180A / 32768 = 5.5mA is the smallest current step that SBMS0 can measure that is 0.005A and so good enough resolution to measure very small loads fairly accurately.

Even with a 75 or 50mV shunt you can still see a 10mA change in current and even that is more than sufficient for small loads even those in the mW range.

Question: If I were to exceed the range of the SMBS0 on the high side on rare occasion and for short durations, what data will the SBMS0 use and log? If I ran 205A thru a100mV 200A/ shunt for 10 seconds, will the SBMS0 simply think that it's 180.22A and I introduce a tiny SOC inaccuracy or will something else happen?

Answer: Not answered.

2.5 Shunt Calibration with 50, 75, and 100mV Shunts

The shunt settings are found on the

Main Menu / Device settings / Automation / Ext ADC / Ext Battery Shunt

and / PV 1 Shunt

and / PV2 Shunt

Our shunts are all 100A / 100mV, so the correct setting in the Shunt page is 1mOhm.

<email discussion between Jon Hacking and Dacian January 17, 2021 (paraphrased)>

The SBMS0 zero-calibrates the shunts on power-up, so it's important that no current is flowing after you connect your shunts, and after hooking up your shunt(s), you must then cycle power on the SBMS0.

If the SMBS0 is controlling loads and charging sources as designed, all loads and all sources will be turned off at power up, so this happens automatically. But if you power up the SBMS0 and it is NOT controlling the loads and/or charging sources, if current is flowing on power up, your current values will be wrong, and balancing may be affected (because you need at least 300mA flowing for balancing to occur).

The automatic zero-calibration values for the shunts are shown on the Diagnostics page.

Question: Have you noticed the cheap shunts "drifting" over time? I had emailed you once before when the SBMS was showing a -1A load when everything was off & disconnected. That was solved by unplugging the BMS ribbon cable and reconnecting, and everything was fine for a good while. But now I am seeing a 0.2-0.3A draw on the BMS even with everything disconnected. I could reset the BMS again, but wondered if this is something possibly caused by the shunt.

Answer: Shunts can not drift over time not in the sense that the zero current point will move. Are you sure that everything is disconnected ? A photo of the first monitoring screen and one of the diagnostic screen when current through the shunt is zero may be helpful.

2.6 Shunt Zero Calibration

<From Dacian 06-Mar-2021>

Note: This is referring to the SBMS40, but I believe the comments about the method applies to all SBMS models.

Load current is not measured but calculated as the difference between battery current and PV current so if the PV and battery current shunts are not properly calibrated you may have situation where you will see a fake Load current and the Load current will fluctuate with the PV current.

The way you will calibrate is by having a known PV current and no load at all. This way, the current through the PV shunt and battery shunt will need to be the same as they are in series. If there is any difference between this two it will be calculated as Load current.

So say PV shunt reads 10% higher than battery shunt and say real current is 10A but PV shunt read 11A then battery shunt say it reads correct 10A then SBMS will think that there is a load 11A - 10A = 1A and if current increases to 20A real then PV shunt will now read same 10% higher at 22A while battery shunt will still read correct 20A now Load will be calculated as 22A - 20A = 2A thus it will appear like a Load increasing from 1A to 2A while there is actually nothing connected to Load.

So you will typically calibrate the battery shunt as that is bidirectional and has better zero offset auto calibration then after that is calibrated based off a known reference say a good multi meter then you can calibrate the PV shunt based on the battery shunt (adjust PV shunt resistance setting so that it reads the same current as the shunt).

2.7 Fuses and Switches

The SBMS0 requires the shunts to remain connected to the battery at all times to avoid inductive disconnection transients from damaging the SBMS0 inputs - i.e. protection devices must be installed "after" the shunts, not between the shunts and battery.

Question: Can a 600A battery disconnect switch be used between the battery and shunts? I currently have 2 disconnect switches - one after each shunt, but it's probably not valuable to have two separate switches.

Answer: No you cannot have anything that can disconnect the battery from the shunt including a disconnect switch. That should also be after the shunts and yes you either need two of them or one with dual separate contacts as you will not want load and chargers to be connected together without a battery since they can get damaged.

Q: If I remember an old discussion, we cannot put a breaker or equivalent to disconnect battery to shunt but we can put a breaker on the negative battery side right?

A: You will not want to fully disconnect the battery from the circuit no matter if is on the negative or positive side. You will need to have a switch for Loads and a separate isolated switch for charge sources (the two separate switches can be connected to a single physical switch handle) but you will not want the charger's to feed the loads without the battery in the circuit as that can damage the loads or and chargers.

Question: A switch on the negative seems like a good solution. Is there any danger in having the positive connections live if the negative is disconnected?

Answer: It depends. In a vehicle the body of the car is connected to negative so if your LiFePO4 battery is also connected to body of the car the positive wire can still produce a current if shorted to the body of the car.

Also see above you need separate switch circuit for loads and charge sources and there is no advantage on having that on the negative side as opposed of positive side as shown in the SBMS0 user manual in the main diagram where there is a circuit breaker for loads and one for charge sources and so those can be used to disconnect the battery from the circuit.

3 Saving Parameters and Device Settings

<google group discussion "Parameter Storage" December 30, 2020 (paraphrased)>

Question: I unplugged the SBMS0 to do some wiring work, nothing to do with the ribbon cable and battery sense, now when plugging it in all the parameter settings are gone!! Have spent the last 2 hours re-entering them, trying to save and unplugging the ribbon cable to only have the same result, told it will need a reset etc. Prior to doing this the settings were saved.

Answer: The ones on the Parameter Settings page are saved in the ILS94203 and you should not have anything to do in advanced parameter settings just battery type number of cells in series and capacity then Store Parameters and power cycle for them to take effect.

All other settings in Device Settings and Automation are be saved permanently by pushing the Save Device settings button.

Our own notes Feb 2021: To change any parameter on the Parameter Settings page or Advanced Parameters page, you need to go back to the Parameters Page and press "Store Parameters". This stores the settings change in the EPROM, but does not yet set it in the BMS chip. In this state, the BMS won't balance (I think).

So you need to cycle power on the BMS to get whatever that flag is cleared and the updated parameters saved in the BMS chip.

The correct way to do this is (after “save parameters”):

1. Turn off all ins and outs to the battery (the big blue seas switches)
2. Remove the ribbon cable on the BMS, wait 5 seconds, and reconnect. (ie cycle power)

If you don't do this after changing Parameter Settings or Advanced Parameter Settings, the Electrodacus won't balance.

When you do power cycle the Electrodacus, it loses your state of charge, and the time/date, and it also re-calibrates the shunts (why you want all current flow off).

So after power cycling, you need to do the following:

1. Go into Device Settings, press set key twice, and you can change the date/time. Then Save Device settings (on the first page).
2. To reset the SOC, you have to run the voltage up to 14.2 to get the Electrodacus to initialize the State of Charge to 100%, then it uses the measured current flow into and out of the battery to keep track of the SOC (NOT the battery voltage).

4 Balancing

<google groups discussion “Better battery cell wiring...” comment on January 1, 2021>

The SBMS0 is based on the Intersil ILS94203 and this will balance as soon as there is a minimum of 300mA of charge current and as soon as there is more than 10mV delta between any two cells thus it can balance for many hours each day if needed and in total it can handle way more energy due to long balancing time even if at low currents. Balancing 30 seconds at 1A is the same as balancing 5 minutes at 100mA

The type of balancing used by the SBMS0 is used in all commercially available EV's including Tesla.

From < https://github.com/bdlow/sbms0 >

Cell balancing is achieved via a 24 Ω resistor switched across each cell by a FET controlled by the ISL94203; nominal balancing current is therefore 3.6 V / 24 Ω = 150 mA. Cell balancing is only enabled when:

- all cell voltages exceed 3.2V, and
- cell voltage deviation exceeds 10 mV, and
- charging current exceeds 300 mA

These parameters are configurable in the ISL94203; refer the datasheet for details.

<email conversation between Jon Hacking and Dacian Jan 17, 2021>

Question: A setting in the SBMS0 (that I’ve left at the default) says that it should balance for 3 seconds & then measure for 3 seconds. Is there any reason why I shouldn’t change this to 3 secs balancing & 1 sec measuring?

Answer: You will likely not be allowed to do that as the cell balancing needs a minimum time OFF so average power on the cell balancing resistor is not too high.

Question: If the balancing is simply to put a load on that cell to bring it down, how many cells will it balance at a time if one cell is low?

Answer: All cells minus the low one can be balanced at the same time. There will actually be no load on any cell as cell balancing only happens if there is at least a 300mA of charge current. So say you have just the minimum 300A charge current and our cell balancing is around 120mA then the low cell will receive 300mA charge current while all the others will just recive 180mA of charge current so all cells will be charged--but the higher cells will receive a lower charge current, as some of that current will go through cell balancing resistors.

Question: I still don't understand about the balancing On/Off times. Are you saying that it can only tolerate a 50% duty-cycle on balancing? That the dump resistor(s) could get too hot with a higher duty-cycle? That would be unfortunate, as it would mean that it will take twice as long to get a given amount of balancing in.

Answer: I made a test with my 100Ah 8s 24V LIFePO4 GBS battery by having it perfectly balanced to start with and disabled the cell balancing for 4 months. After 4 months the total cell imbalance was 2Ah that means that when fully charged one cell was full and the lowest cell in the pack required 2Ah to be put in to also be fully charged. The battery was used every day to power my house so charged from solar PV and discharged every day for what the house needed.

This will be an average of around 0.5Ah per month or 0.016Ah per day and that means that cell balancing will have been needed an average of just 5 to 6 minutes per day to keep this 100Ah battery balanced. Thus batteries with capacity of 2000 to 3000Ah can be kept easily in balance in a solar offgrid system as there are at least a few hours of solar charging each day when cell balancing can be performed if needed.

Because of this all BMS ICs use passive cell balancing because active cell balancing will be way too expensive, and never be able to recover the investment.

For example that 100Ah 24V battery required 0.5Ah x 25V = 12.5Wh/month x 12 months = 150Wh and over 10 years 1.5kWh. So the total loss in 10 years due to cell balancing will be 1.5kWh. In the case of the SBMS0, all this cell balancing current will come from PV panels and those cost around 2 cent/kWh so a total of 3 cent worth of energy was wasted in 10 years.

You can see how no complex circuit like a DC-DC converter can make any sense for any installation--even with a very large 2000Ah battery, the cell balancing energy will be 20x larger. Still, that will be 60 cents total value over 10 years.

5 Input / Output Connections

An especially useful summary of I/O's for Electrodacus is located here:

https://diysolarforum.com/resources/electrodacus-sbms0-i-o-summary-chart.112/

This chart was made up by a guy who actually owns an Electrodacus (latest version as of early 2021) and so should supersede anything below, if they conflict.

The below info came from this source: https://github.com/bdlow/sbms0 . I don't think this guy ever updated it after he used the BMS (if he did in fact use it). It appears this is just research, and may not actually be factual. My guess he was taking notes while trying to figure out which BMS to buy, and eventually bought something else.>

References

- https://groups.google.com/g/electrodacus/c/gsGjTV3cbC0/m/JS0Mkci2AAAJ

- https://groups.google.com/g/electrodacus/c/2OB3qrNVyYU/m/HE2ZlcKOBAAJ
- Put simply, the SBMS0 has two relatively protected voltage measuring inputs for the load and charging shunts and one resistance measuring input for a temperature sensor, and multiple digital outputs that act as low-current switches. The digital outputs *must* have external current limiting (typically, a resistor).
- In more detail: all IO connectors are differential: they have separate positive and negative connections, as opposed to a single-ended IO with a common/shared ground. This is advantageous for better noise rejection and also ensures all connections are isolated including from each other (up to 76V for the analogue inputs). The ADC and PV inputs are directly connected to current-sense amplifier sense inputs; the ADC is bidirectional - can measure current flow in both directions - and the PV input is unidirectional (able to measure positive values only, as charge current normally only flows from PV to battery). The latter has some implications with respect to calibrating any measurement offset errors, discussed in shunt calibration, below.
- The optocoupled digital outputs are isolated Darlington transistor outputs and essentially behave as a switch that can sink up to a *maximum* of 150 mA - under 50mA is recommended by ElectroDacus (ref).
- The mid-2020 v03d revision SBMS0 switched to using TLP172GM photorelays which are bidirectional (polarity doesn't matter) but otherwise functionally equivalent (maximum 110mA).
- The outputs **must** be externally current limited - connecting the output directly across power and ground will result in a failed optocoupler. Though many controlled devices, such as a BatteryProtect, incorporate current limiting, if any part of the switched circuit is connected to battery positive then it's probably a good idea to include a 1-10kΩ resistor in series.
- The outputs can be reconfigured to suit multiple functions, described in the next section. The default configuration is given in the following table:

Pin Label	Function	Description	Technical Note
ADC	Analog Input	Primary (battery) shunt	current sense input
PV	Analog Input	Charge (PV) shunt (optional)	current sense input
XT	Analog Input	Thermocouple input (optional)	10kΩ input impedance
EXTIO3	Digital Output	Default: load on/off	optocoupled
EXTIO4	Digital Output	Default: charge on/off	optocoupled
EXTIO5	Digital Output	Default: unused	optocoupled
EXTIO6	Digital Output	Default: 'fault'	optocoupled

- EXTIO1 and EXTIO2 are skipped (used for isolated data I2C).

Each of the four EXTIO outputs can be set to one of six different modes or "types".

The SoC-based outputs (Type 3 and 4) should not be used for ensuring battery safety - a State of Charge measurement is not sufficiently accurate to reliably protect the battery under all conditions. For example, multiple partial charge-discharge cycles will cause SoC integration errors to accumulate and will result in an inaccurate SoC measurement

5.1 Ext I/O's

5.1.1 Type 0 unused

- an EXTIO set to type 0 is effectively disabled

5.1.2 Type 1 HVD - High Voltage Disconnect

- charging control
- voltage sense: disconnects when battery reaches (TODO, adjustable?) xx V
- used to control any charger that can be DSSR20, an MPPT solar charger with remote ON/OFF, a grid charger or a battery to battery charger

5.1.3 Type 2 LVD - Low Voltage Disconnect

- load disconnect
- voltage sense: disconnects when battery reaches (TODO, adjustable?) xx V
- control remote on/off for load devices such as an inverter or Victron Battery Protect

5.1.4 Type 3 High SoC Alarm

- per Type 1, however based on State of Charge

5.1.5 Type 4 Low SoC Alarm

- per Type 2, however based on State of Charge
- may be useful to switch an "optional" load, for example disabling an air conditioning system when estimated SoC falls below 50%

5.1.6 Type 5 Fault

- secondary high or low voltage limits (under voltage or over voltage lock) reached
- can be used as backup in the event a controlled charger or load fails to respond
- should be connected to a buzzer or similar alerting mechanism, and/or a remote triggered circuit breaker or similar device (the latter are not easy to come by)

5.1.7 Type 6 Dual PV Control

- switch an optional additional charging source
- type 1 (primary) and type 6 (secondary) outputs are controlled together such that the total charge current from both sources is maximised but always lower than the configured battery limit (Imax)
- the two charging sources (for example, two PV arrays) are separately controlled: one via an EXTIO set as the standard type 1 output, the second EXTIO with this type 6 output

- the type 6 charge source is assumed to be around twice the capacity as that of the type 1 charge source; e.g. a 6 panel PV array as type 1 and 12 panel PV array as type 6
    - in other words, the secondary source is assumed to contribute twice the charge current than the primary source
- the control is stateful: the SBMS0 will only proceed to Level 3 (both sources) from Level 2 (secondary source only), similarly to Level 1 from Level 2
    - the "going down" threshold is set at 120% of Imax, i.e. charge level is reduced when charge current exceeds 120% Imax
    - the type 1 (primary) source is presumed to be matched to the battery and not able to exceed the battery charge current limit

Level	Type 1 output	Type 6 output	SBMS0 Display
1	enabled	disabled	L1
2	disabled	enabled	L2
3	enabled	enabled	L3

Question: Is there any prospect there might be expansion of the automation settings in the Electrodacus firmware in future, to give users more flexibility in controlling external devices via the EXTIO ports?

A couple of scenarios I'd like to make use of don't appear to be covered adequately in the existing type 1- type 6 automation settings.

a) Switching a "fall back" or "emergency" charging source when battery SOC falls to a user defined level, with disconnect of that output at a further user defined level. Eg switch on mains connected switch mode PSU for emergency charging if SOC falls to 10% and switch off again at 30%

b) Providing a momentary pulse of user defined duration, which can be triggered by exit ADC 2 or ext ADC 3 reaching a certain level. Here I'm thinking that I could feed one of my EXT ADC ports my PV panel array voltage via a potential divider (to stay within the 1.8V limit) and trigger a momentary closing contact output on one of the EXTIO ports to switch on a charge controller which needs a button press to begin it.

c) It would also be good to see some of the existing type1-type 4 automation options allow specific cell voltage thresholds to be set, rather than relying only on the calculated SOC level (which can be wrong in some circumstances).

Answer:

a) Something like this already exists is the type 3 and you can set say a 20% SOC limit that way that charger will only enable when SOC drops 3% below that set limit and stops at 20% keeping the SOC there. But if you have a grid connection and battery is low you should just move the loads to the grid it will not make sense to involve the battery.

b) I may add some automation related to ADC2 and ADC3 if they exceed or fall below user set levels. If you have the SBMS0, the ADC2 and ADC3 already have built-in resistor divider allowing for 0 to 60Vdc measurements. The momentary push button devices cannot be used as you do not know the state of the device and you just toggle the last state so SBMS will not know if by providing a pulse it turned ON or OFF that device. On top of that, it is not safe to use as if EXT IOx connection is lost for any reason the device (charger or load) will remain in the last state instead of safely turning OFF as it is the case with close circuit for ON and open circuit for OFF.

There is somewhere on this forum a post on a simple schematic that can convert a device with momentary switch to be compatible with open or close signal from the EXT IOx, but for this to work you need the push button signals and the ON LED as confirmation that device is ON or OFF.

The type 1 and type 2 are not based on SOC, they are based on cell voltage levels--the Over voltage threshold and the under voltage threshold.

The type 3 and type 4 were mostly intended for alarms and they are based on SOC but the cell voltage will still overrun the decision of SOC. So say you set a load (maybe an inverter) as type 4 and want that to turn OFF at 30% SOC and say for some reason SOC calculation is wrong and shows say 60% but battery is empty (one of the cells is below undervoltage say 2.8V for LiFePO4) then even if SOC is at 60% and type 4 is set for 30% the 2.8V low cell voltage will overrun that setting and inverter will still be turned OFF.

Question: So if I understood correctly, with a type 3 EXTIO port set to 20%, the contact is open circuit until the SOC declines to 17%, then it closes and assuming the SOC increases from there, the EXTIO port goes open circuit when SOC is up to 20%? Is this 3% hysteresis setting adjustable or fixed?

A momentary contact output can be a reliable way to control an external device, depending how that device reacts to the contact. We can't assume that users will only use controlled devices that have more than one function activated from the same momentary switch. The icharger which I would like to trigger with a momentary contact applied to the "start" button will run a charge program then terminate safely without needing to receive further commands.

I realize there are limits in the control you can offer via SBMS units, but I do think the automation features differentiate your product from many BMSs and there's scope to develop the functionality offered, if you wish to do so.

5.2 Temperature Sensors

<email discussion between Jon Hacking and Dacian January 17, 2021 (paraphrased)>

A battery temperature sensor is not required, however, if you don't have a temperature sensor, do not set limits for low and high battery temperature. This will cause the SMS0 to disable charge and discharge, and balancing will also be disabled.

The default parameters should be this, if you are not using an external temperature sensor.

Min Ext Temperature 4096b

Max Ext Temperature 0000b

This way the external temperature sensor is disabled and you charge and discharge are enabled thus also the cell balancing.

6 Disconnecting Charges and Loads

6.1 DSSR20 Solid State Relay

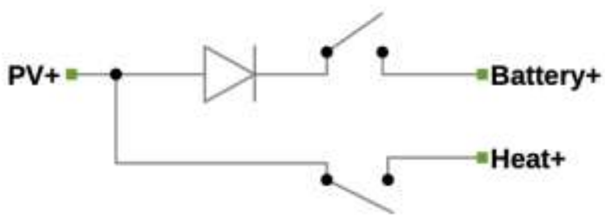
Source: This Section 8.1 comes from this source: https://github.com/bdlow/sbms0 . I don't think this guy ever updated it after he used the BMS (if he did in fact use it), this is just research, and may not actually be factual.

PV Input fusing is not technically required as the DSSR acts as a diode, no current can flow from the battery to the PV connection beyond the DSSR. A fault in the DSSR (e.g. MOSFET short circuit) could impact this diode behavior, so it is prudent to incorporate a fuse. Fusing is required for an external charger that is either uncontrolled, or by something like a Victron BatteryProtect, so it makes sense to use the same fuse for both.

As with the SBMS0, the DSSR20 is occasionally updated. The following pertains to the DSSR20 v04d, ca. late 2019.

The DSSR20 is a low-cost solid state switch with one or two twists. The first twist is the presence of an "ideal" (low-loss) diode in the PV-Battery path: this is advantageous in terms of mitigating certain types of failures that can occur on the PV side such as cabling shorts. The second twist is that it is available in a version that adds facility to switch a secondary resistive load - very useful for "heating diversion" as described above.

The DSSR20 functional diagram is as follows:



The Heat diversion path is not protected by an ideal diode, and is not designed to connect to an "active load" - one that can source current back to the PV. i.e. do not connect a power source such as a battery to this connection; passive loads only. Further, the load is not rated for reactive (inductive or capacitive) loads, so no motors or inverters. The Heat diversion control input is separate to the Battery control input; in the intended use case they would be logically inverted. i.e. when Battery is on, Heat is off and vice-versa. The DEXT16 module implements this logic given a single EXTIO output from the SBMS0. The DEXT16 also provides screw terminals and can make for a tidier installation when multiple DSSR20s are used.

The DSSR20 is powered from the PV input with a voltage monitoring reset IC that ensures it only operates when sufficient voltage is available. The MOSFET switches are off / open when the DSSR20 is not powered.

The DSSR20 is protected against reverse PV polarity, both the DSSR20 itself and the Battery and Heat outputs.

Power dissipation should be less than 2W under all operating conditions, and no heatsinking or forced cooling is required.

6.1.1 The Ideal Diode and Fusing

When used as intended, no fuses are required between the PV panel(s), DSSR20(s) and battery. Eliminating fuses is a benefit in terms of cost and reliability.

The diode in the PV-to-Battery path is necessary to protect the DSSR20 in the event of a short circuit on the PV side. With only a battery fuse and no diode, a PV-side fault would see very high currents delivered from the battery until the battery-side fuse blew - no fuse can react quickly enough to prevent damage to the MOSFET switch. The ideal diode blocks any such fault currents and unlike a standard diode does so with very low power loss.

Note the diode does nothing to protect the wiring between the battery and DSSR20: if there is any chance of a fault between the battery and DSSR0 - for example, if they are not located in the same protected enclosure - then a fuse must be installed near the battery to protect the wiring between the battery and DSSR20.

The DSSR20 is designed to handle up to two commonly available panels - typically around 10 A each, with a maximum of 20 A via the Battery or Heat outputs. With only one panel no PV-side fuse is required as the panel cannot possibly exceed its rated limits (it cannot deliver more than short-circuit current, Isc). With two panels in parallel it's still likely that no fuses are required, however it would be safest to check the manufacturer's ratings to be certain (alas, usually not specified). More than two panels in parallel creates more complex fusing requirements and is not recommended; each DSSR20 is intended for one or two panels to keep things simple.

6.1.2 Non-Standard Configurations

The DSSR20 can handle 20A via the Battery output and 10A (?TBC) via the Heat output. Multiple DSSR20's can be placed in parallel to increase current capacity (internally, the PV to Battery+ path is two pairs of MOSFETs in parallel controlled by one IC, a parallel DSSR20 essentially adds another two pair in parallel - though with the distinction that a loose wire could cause the control signals to not be in sync and thus give rise to an overload of one of the DSSRs). The practical limit - before imbalances due to wiring resistances, etc become a problem - would be two DSSR20's. If for your use case you need to carry more than 20A, it may be more economical to use a Victron BatteryProtect instead of two DSSR20's - though note the BatteryProtect does not incorporate an ideal diode, so appropriate fusing is essential and even then any shorts on the PV side may toast the BatteryProtect as it is not designed to handle reverse currents.

Fusing on the PV side may still be required depending on the panel topology and characteristics, for example if two panels are used and they are not rated to handle the maximum combined current, a fuse on each panel should be used at the panel combiner junction.

6.1.3 Using DSSR20's with 24v Panels

Question: We have 4 24v nominal solar panels (200 Watts each) with a Morningstar MPPT Solar Controller. If two of the DSSR20s can handle up to about 25a (12.5a each) from the 4x200watt panels at 24v in and out of the unit (50a at 12v after the MPPT to the batts) that would work perfect for us. Then later, if needed, the charge buss bar HVC would cut batt power to the MPPT, but the panels would no longer be connected. Will this work?

Answer: If you can split the PV array in two 2x200W + 2x200W then you can use two DSSR20 one for each 2x200W. If panels are already all parallel and it is too difficult to split the array in two, then you can parallel two DSSR20 as is not far from what a single DSSR20 can handle. Those likely are 72 cell panels with 5" cells.

6.2 Other Disconnect Devices

Question: Can I use the SBMS0 with disconnect devices other than the DSSR20?

Answer: Any device that can disconnect current in response to an SBMS0 EXTIO output can be used. e.g. a solar controller (if you insist on using MPPT, you can!), mains or DC-DC charger, or some other solid state or electromechanical relay (for a electromechanical relay, ensure a flyback diode is installed).

Latching relays are not usable for either LVC or HVC. If a remote signal is broken the relay (or remote ON/OFF device) should default to OFF for safety and a latching relay will usually remain in the last state so not safe.

People for small DC loads use the Victron BP65, BP100 or BP220 which are solid state relays and can be controlled by the SBMS0 directly.

6.2.1 Latching Relays are Not Recommended

Note: We originally bought latching relays because they only use current when you change their state, and then they stay in that state until you direct them to change state (closed to open or open to closed). Dacian strongly disagrees that this is the right thing to do. His philosophy is that if the Electrodacus dies or is shut down, all the relays should go open, disconnecting both loads and charging sources from the battery. So he recommended "normally open" solid state relays.

Question: (from Dave McCampbell email exchange Nov 2020): Regarding the BlueSea 7700 latching relays we had planned to use for our bus cutoffs, are you saying there are two problems with them:

1 danger from signal wire being broken/disconnected

2 that the relays cannot be controlled by the SBMS0

Does this mean we absolutely cannot use them with the SBMS0 and need to buy the Victron Battery Protect devices?

Answer: You want a relay that will default open. This is the safest way. There are multiple BlueSeas relays, some will have the correct type of remote where when the remote signal is removed they will automatically turn OFF (open circuit).

For example the 7713 or 7717 http://assets.bluesea.com/files/resources/instructions/980018140-002.pdf will work as it will default to OFF when the remote signal is removed but still needs 4A for 20ms when changing state so you need a CPC1706 in order to control that.

I'm not that familiar with the BlueSea relays. The SBMS0 EXT IOx are open or close circuit signals with max 50mA capability. So when circuit is open the relay will also need to be open and as long as the circuit is close circuit the relay output will also need to be close circuit. This way if signal wires are broken/disconnected, the relay should automatically default to open circuit so OFF and it seems the 7700 uses two separate momentary switch signals not compatible with SBMS0, and I do not consider that safe.

6.3 Battery Disconnect with PV Still Connected

Question: Can the SBMS0 survive a battery disconnection when PV is still connected?

Answer: Yes. Some solar controllers that directly control a PV source can fail when the battery is disconnected with the PV still connected. The Electrodacus SBMS40/60/100/120 models, incorporating the switching electronics in the control unit, are susceptible to such a failure mode - those units must first have both PV and load removed before they can be disconnected from the battery.

The SBMS0 is designed such that it is safe to disconnect the cell monitor cable at any time (doing so will also safely turn off the controlled DSSRs and loads). However it is not safe to disconnect the battery from the SBMS0 shunt when any current is flowing (charge or discharge). If the battery is disconnected under charge or load a voltage spike will occur due to inductance in cabling and loads. This spike can damage the SBMS0. As such fuses and circuit breakers should be connected "downstream" of the SBMS0's shunt.

7 Wi-Fi

The Wifi module is an optional add-on, that adds an ESP32 that can receive data from the Electrodacus and transmit that data via wifi.

7.1 Wifi Setup

7.2 Resetting Access Point Credentials

Question: Once provisioned as a client to a LAN, can I force the ESP32 back to it's softAP default without LAN access?

My scenario:

- Provision to a Router/SSID as client in one location
- Move the SBMS0 to a new location with new Router/SSID without reset of prior credentials

Answer: Not sure but I think there is no way to reset to AP. Robert may know better but I think the only fix is to connect to the old router or just flash the ESP32 with the firmware as that should delete all settings and return to default.

7.3 Wifi is Transmit-Only

You can only have a single device connected to the SBMS.

SBMS cannot serve multiple devices as it ignores all requests and only sends one set of data every second.

Note: sometime in mid-2020, another guy took over the code for the wifi-add-on. The comments below may be before that time, and may not still be true.

https://github.com/armageddon421/electrodacus-esp32

The AT-firmware that runs off the ESP32 is fairly beta version likely fort very robust to multiple requests.

I only control the software written in the SBMS main micro controller that is real time and cannot fail the AT-firmware written by Espressif is not real time so if is overwhelmed with requests it may fill some buffer or get stuck.

The way I implemented the WiFi is transmit-only so completely non-standard in order to be able to protect the main microcontroller from denial of service type attacks so microcontroller will not respond to requests in any way and only sends the data once a second thus one device can receive that data but not multiple devices.

As this works and micro controller cannot lock (do not have the SBMS on the WiFi screen as that is also used as diagnostic so more unnecessary communication with ESP happens so after you reset and turn WiFi off just get out of the WiFi menu).

It is clear that the firmware on the ESP32 fails likely when there is too much demand as I'm only connecting occasionally to take a screenshot to either SBMS120 or SBMS0 and they are both connected for many months with WiFi communication never failing.

The only other variable is that I use my desktop to connect and this runs a Linux distribution but so will a Raspberry Pi.

So I'm only connected maybe for one minute then disconnect as I need to connect to internet.

Ideally a better firmware for ESP32 should be written I just did not had the time (never a priority) and the no so open HW and SW of the ESP32 makes it non attractive to me.

7.4 Wifi and MQTT

Browse this thread

https://groups.google.com/g/electrodacus/c/8eWBUiigJUQ/m/UXt3Q5MkBAAJ

and here: https://github.com/armageddon421/electrodacus-esp32

From https://github.com/armageddon421/electrodacus-esp32

What is MQTT?

MQTT is a publisher-subscriber based message passing protocol that uses a central server called "broker" to relay the messages. Many IoT applications nowadays support MQTT, devices as well as dashboard and home automation solutions. It is easy to locally host your own local MQTT-Broker (for example Mosquitto) for your home automation needs, a raspberry pi is already enough. I personally use node-red as "glue logic" to connect all my sensors and actors, feed my databases and for creating logic.

From MQQT Payload Documentation – Feb 14, 2021

For anyone wondering about the MQTT payload object, I've pasted it below from the Node-RED debugger for others to reference.

Everything is pretty self-evident and reflective of the internal data logging output (page 20-21)

Below this is my visualization sandbox in preparation for the new house...

I've simply powered up my unit for evaluation with a resistor divider to battery simulation so I could get comfy with decode and display of dashboard data.

The SBMS0 payloads visualization sample is in the leftmost "off-grid" panel

object

time: *object*

    year: 0

    month: 1

    day: 1

    hour: 4

minute: 2

second: 55

soc: 49

cellsMV: *array[8]*

0: 2905

1: 2916

2: 0

3: 0

4: 0

5: 0

6: 2924

7: 2935

tempInt: 28.6

tempExt: -45

currentMA: *object*

battery:-1100

pv1:0

pv2:0

extLoad:1040

ad2: 0

ad3: 0

ad4: 0

heat1: 0

heat2: 10774

flags: *object*

OV: false

OVLK: false

UV: false

UVLK: false

IOT: false

COC: false

DOC: false

DSC: false

CELF: false

OPEN: false

LVC: false

ECCF: false

CFET: true

EOC: false

DFET: true

7.5 Programming the ESP32

7.5.1 Programming using the USB Cable

<Dacian on [Google Group Feb 8, 2021]>

The WiFi (ESP32) firmware is a separate thing from the SBMS0 firmware.

To program the ESP32 on the new SBMS0 all you need is an USB cable and you need to flip some switches to put it in to program mode then back in to normal working mode. I do not think Robert has any newer version of firmware compared to the one that already comes with the SBMS0.

You can use the html file as transmitted by the ESP32 or you can download the html file on your device and edit that html file with the link where to get the raw data then you can modify the html file the way you want.

Not sure I answered your question as I'm not sure what exactly you are asking.

<From [Robert's Github Page]>

If your SBMS does not come pre-flashed with an OTA capable firmware version yet, you need a USB to Serial/UART converter with an interface level of 3.3V. 5V will destroy your ESP32. And any "PC" with any OS supported by plattformIO with installed python (2.7 should do). Make sure pip works.

The basic procedure is as follows:

Disconnect the WiFi shield with the ESP32 module from your SBMS. To be safe, unplug the SBMS completely before doing that.

Connect the pins RX,TX,GND and 3.3V as shown in the illustration. Your USB to Serial converter will probably already provide 3.3V and enough power, so you can use that. The RX of your converter must be connected to the pin labeled TX and vice versa. It is probably easiest to use female jumper wires.

Make a temporary connection (hold a male jumper wire onto it) from the pin labelled IO0 to GND. This will bring the ESP32 into bootloader mode so you can flash the new firmware.

Only now plug in your USB to Serial converter to your USB port (or power up your dedicated 3.3V source). You can release the connection to IO0 now.

Flash the firmware (see next section)

Verify you can reach the device via WiFi. It should create an Access Point with a name starting with SBMS, followed by 12 characters representing a MAC-Address. The default password for that AP is "electrodacus" (without the quotes). Connect to it by going to 192.168.4.1 in your browser.

Disconnect everything you just connected before plugging the WiFi module back onto your SBMS.

7.5.2 Doing an On The Air Update of the ESP32 Firmware

<Robert on Google Group Nov 2020>

As you can still connect to the wifi of the esp32, you may try updating the firmware over the preliminary, not really well tested web update form.

Go to /update on your sbms and then upload first the firmware.bin and then the spiffs.bin from https://github.com/armageddon421/electrodacus-esp32/releases

<From Robert's Github Page>

ArduinoOTA procedure

Make sure you are connected to the same WiFi as your SBMS, or are directly connected to its own WiFi. Flash the firmware (see next section). In sections 4 and 6, add -e ota --upload-port [IP/hostname] to the end of the commands. If you are directly connected to the SBMS' own WiFi, the IP will be 192.168.4.1.

7.5.3 Installation / Flashing

Open a terminal or cmd window.

1. Clone the repository by either downloading the zip file manually or running:

```
git clone https://github.com/armageddon421/electrodacus-esp32.git
```

And enter the directory

```
cd electrodacus-esp32
```

2. Install PlatformIO

```
pip install -U platformio
platformio update
```

3. Compile the code. Platformio will automatically install any missing dependancies. Check for any errors.

```
platformio run
```

If all goes well, you should see memory usage bars like these and a line below that reading [SUCCESS] in the middle.

```
RAM:   [=         ]  13.5% (used 44224 bytes from 327680 bytes)
Flash: [=======   ]  74.3% (used 973461 bytes from 1310720 bytes)
```

4. Flash! Platformio will usually find the right serial port to use automatically.

```
platformio run --target upload
```

Note: For OTA updates add

`-e ota --upload-port [IP/hostname]` to the end of the commands.

5. After the process is finished, replug your USB (or power supply) to reboot the ESP32 and make sure the GPIO0 is still connected to GND so you can flash again.

Upload the filesystem

```
platformio run --target uploadfs
```

6. After the process is finished, disconnect GPIO0 and replug your USB (or power supply) to reboot the ESP32. That should be it.

7.6 Decoding the Data on the ESP32

<Dacian on Google Group Feb 8, 2021>

The SBMS0 firmware is separate as mentioned and written for the STM32F373 that is on the main SBMS0 board and separated from the WiFi firmware.

The SBMS0 will transmit that compressed data over UART to the ESP32 or if you want to the USB port and that data has already a fixed format documented in the user manual.

If data was not compressed in base 91 then data will be at least 2.5x larger in size. Base 91 is no different form base 16 (Hex) or base 10 (Dec) and the formula and all details on how to decompress is present in SBMS0 user manual.

Below an answer I provided recently to someone over email related to base 91

He asked about the USB data and what it means and sent me this photo with a few lines of data thinking data is wrong.

```
8,0/)@#TIJH{########I@I6)h##-##3#####################$?e&M*
8,0/)^#TIJHz########I@I6)a##-##2#####################$?z&M*
8,0/*@#TIKHz########IAI5)d##-##2#####################$?a&M*
8,0/*^#TIKHz########IAI6)h##-##3#####################$@*&M*
8,0/+@#TIJHz########I@I5)h##-##2#####################$?d&M*
8,0/+^#TIJHy########I@I5)k##-##4#####################$@+&M*
8,0/,@#TIJHz########I?I5)h##-##1#####################$?c&M*
8,0/,^#TIJHy########I@I5)d##-##3#####################$@$&M*
8,0/-@#TIIHy########I@I5)d##-##4#####################$?b&M*
8,0/-^#TIIHz########I?I5)f##-##5#####################$@(&M*
8,0/.@#TIIHz########I@I5)a##-##4#####################$?a&M*
8,0/.^#TIIHy########I?I5)f##-##3#####################$@&&M*
```

The data seems correct. What you get over serial is the sbms variable and the content of that is explained at page 20 in the manual.

For example, that first character "8" represents the year just compressed using that base 91 simple compression.

I use the ascii characters for 35 (#) up to 126 (~) so a total of 91 characters.

Thus "8" is number 56 and so 56 - 35 = 21 meaning year 2021

Next character is for month "," and that is 44 - 35 = 9 (not sure why you set month as september).

The data log seems to be done every 30 seconds

Then "#T" is the state of charge and since # is zero it is simple to calculate as "T" is 84 - 35 = 49 so most likely you just connected the SBMS as it always defaults at 49% at power up.

Then the first cell voltage is "IJ"   "I" is 73 - 35 = 38 * 90 = 3420 and to that you add "J" 74-35 = 39 so 3420 + 39 = 3459mV so likely $LiFePO_4$ and very close to fully charged. "

Standard 7-bit ASCII Table																							theascii.com
Dec	Hex	Oct	Binary	Char	Description	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	0	0	0	NUL	Null character	32	20	40	100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	1	1	1	SOH	Start of header	33	21	41	100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	2	2	10	STX	Start of text	34	22	42	100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	3	3	11	ETX	End of text	35	23	43	100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	4	4	100	EOT	End of transmission	36	24	44	100100	$	68	44	104	1000100	D	100	64	144	1100100	d
5	5	5	101	ENQ	Enquiry	37	25	45	100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	6	6	110	ACK	Acknowledge	38	26	46	100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	7	7	111	BEL	Bell ring	39	27	47	100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	8	10	1000	BS	Backspace	40	28	50	101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	9	11	1001	HT	Horizontal tab	41	29	51	101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	12	1010	LF	Line feed	42	2A	52	101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	13	1011	VT	Vertical tab	43	2B	53	101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	14	1100	FF	Form feed	44	2C	54	101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	15	1101	CR	Carriage return	45	2D	55	101101	–	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	16	1110	SO	Shift out	46	2E	56	101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	17	1111	SI	Shift in	47	2F	57	101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	20	10000	DLE	Data link escape	48	30	60	110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	21	10001	DC1	Device control 1	49	31	61	110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	22	10010	DC2	Device control 2	50	32	62	110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	23	10011	DC3	Device control 3	51	33	63	110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	24	10100	DC4	Device control 4	52	34	64	110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	25	10101	NAK	Negative acknowledge	53	35	65	110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	26	10110	SYN	Synchronize	54	36	66	110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	27	10111	ETB	End transmission block	55	37	67	110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	30	11000	CAN	Cancel	56	38	70	111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	31	11001	EM	End of medium	57	39	71	111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	32	11010	SUB	Substitute	58	3A	72	111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	33	11011	ESC	Escape	59	3B	73	111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	34	11100	FS	File separator	60	3C	74	111100	<	92	5C	134	1011100	\	124	7C	174	1111100	|
29	1D	35	11101	GS	Group separator	61	3D	75	111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	36	11110	RS	Record separator	62	3E	76	111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	37	11111	US	Unit separator	63	3F	77	111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL

<Dacian Google Group Discussion July 27, 2020>

7*)'JX#TGPGTGTGSGTGQGSGL*j##-#1{#######1U################%N(

The 59 ascii characters is all you care about and those will convert in to 27 numerical values as some are just one ascii character so can only represent a number from 0 to 90 like year or minutes then cells voltage is made of two ascii characters and with two you can represent a max value from 0 to 90 * 91 + 90 = 8280 and then with 3 ascii characters you have things like battery current and PV current and there you can represent a value from 0 up to (90 * 91 * 91) + (90 * 91) + 90 = 753570 thus the SBMS0 limitation of 750A (750000mA) when measuring current as more can not be encoded in 3 ascii characters.

The rest of the stuff after ERR will is always just zero. You can see a bunch of ####### and those are only something you will care about if you had a DMPPT450 as they are related to that.

The user manual explains in details what each of the fields are and you only care about the first 59 characters it starts with "7*)" date and ends with "%N(" error code

This is from manual

var sbms="YMDHMS%%C1C2C3C4C5C6C7C8ITET+BATPV1PV2EXTAD2AD3AD4ht1ht2ERR"

Y-year last two digit (limited to 2090).

M-month

D-day

H-hour

M-minute

S-seconds

%% - SOC 0 to 100%

C1 to C8 - cell voltage in mV

IT and ET - internal and external temperature in hundreds of a degree 0 is -45C and 1449 is 99.9C so you need to subtract 450 from the result and divide by 10 (example 689 as result of the decompression will be (689-450)/10=23.9C)

+ - sign for battery charging (+) and discharging (-) used as it is no compression.

BAT - Battery current in mA from -750000mA to +150000mA

PV1 - PV1 input current in mA

PV2 - PV2 input current in mA

EXT - Ext Load current (the use of 10A up to 500A 75mV external current shunt is possible on both SBMS40 and SBMS120 see page 13)

AD2 and AD3 - user available ADC inputs the value is the digital output of the ADC.

ht1, ht2 - reserved for power output of the DMPPT450

ERR - this is error code it will contain the information referring to the error flags on the SBMS.

There are 15 Flags (see monitoring page3 on SBMS) not all flags represent an error some are representing normal operation and if you consider this a 15bit binary number each bit representing one flag with “1” highlighted and “0” not highlighted and you start from the top left with OV flag and continue to right and then down to the last flag DFET and consider the OV as the least significant bit and DFET most significant bit then you get this binary number and convert to decimal that will be the value contained in this ERR Status and you can also see this number on the SBMS LCD just under the battery SOC value and also displayed as Status: in the html page.

Also see this discussion with Dacian June 2020.

https://groups.google.com/g/electrodacus/c/R7vfB0lVzv0/m/uy7j1GQiCAAJ

7.7 Monitoring and Logging and Graphing via Raspberry Pi and USB

Look for David Burton on this thread.

https://groups.google.com/g/electrodacus/c/R7vfB0lVzv0/m/uy7j1GQiCAAJ

I was able to streamline the Raspberry Pi configuration with a script that automated nearly all configurations. There are some initial steps and final steps that need to be done, but I detailed them all in a post here: https://familyburton.com/solar-battery-testing-with-electrodacus-part-2/

Here are the highlights:

- Setup a new Raspberry Pi with Raspian Lite (>= 4Gb micro SD card), steps in the URL

- Run a few commands (they pull a python script from my github repo, https://github.com/Burtond/Electrodacus)

- Import my shared dashboard, also in the GitHub repo, but I highly recommend customizing as you can move graphs around and make it just how you like it :)

*** Don't forget to enable USART logging on your SBMS0, details in the SBMS0 manual,

I was able to test all of the steps on a brand new Raspberry Pi with a 32GB micro SD card, confirmed 2GB was too small, but anything bigger should be fine.

The URL of the Grafana data source needs to be set at the IP address of the Pi. It looked like to was set (at LocalHost), but needs to be changed.

8 User Interface Questions

Below are questions people have asked that are not fully explained in the manual.

8.1 What STAT: 20480 on Bottom Left of Screen Means

<Dacian GoogleGroup July 27, 2020 (paraphrased)>

To convert this number to individual flags, this is a 15-bit binary number and each bit is one of the 15 error flags with OV (OverVoltage) the least significant bit and DFET the most significant bit in the order that are listed in page 3 of the monitoring screen.

From the manual: CFET and **DFET** show if Charge and Discharge FET's are ON or OFF (if highlighted they are ON). Then when highest cell gets to 3.55V (in case of default battery type 1 settings) the CFET will be turned OFF terminating the charging and the OV (Over Voltage) flag will be highlighted but this is normal operation and not an error. Same will be true if battery is getting discharged and the lowest cell gets below 2.8V for more than one second then DFET will be turned OFF and the UV (Under Voltage) flag will be highlighted (again normal operation even if you probably do not want to have an empty battery while off grid).

EOC (End Of Charge) is also a normal operation flag and it will be set when the highest cell gets to 3.53V with battery type 1 default settings.

UVLK or **OVLK** is an error since those limits are set lower and higher than UV and OV and it means that the respective mosfet has failed as a short circuit and can not disconnect the PV or Load or you played with advanced parameter settings and used wrong values in there.

COC, **DOC** and **DSC** are also errors for over-current or short circuit and they can be caused by multiple conditions.

OPEN (open cell wire) is also an error indicating that you have a bad connection to one of the cell monitoring wires while CELF (cell fail) means that one of the cells has a voltage below what is acceptable and may be do to a bad cell to cell power connection or a defective cell with high internal resistance.

LVC is also an error and means that one of the cells has a voltage so low that normal charging is not allowed (you can manually charge that cell with a small current and see if it can be recovered).

9 Environmental Mounting Considerations

Question: I will be installing the SBMS0 in a sailboat. This is quite a humid environment, and I intend to sail to the ocean, i.e. salt water. Do you think it would be a good idea to apply a coat of conformal coating to the pcb, or just an area ?

I have Miller-Stephenson Silicone Resin Coating (Conformal Coating for electronics) MIL-I-46058C Type

Answer: The SBMS0 and all other devices I build are for indoor use. Not sure if conformal coating will help as there are way too many components on multiple boards and connectors so best option in a humid environment is to have the SBMS0 in a sealed box.

Followup Question: If it is a typical problem with the crystal, Is it ok if I just coat the crystal area with a mask ? The sealed box will be hard if I want to use the switches on the panel, but since it will be panel mounted, I can just seal the back.

Answer: As long as there is no water or super high humidity that water condenses on the board the crystal will not be a problem and if there is water on the board the crystal is not the only problem a

I have had a few complaints always in high humidity environments. At some point water may condense inside the SBMS and if water is trapped under the 32kHz crystal that will fail to boot. If you have a soldering iron just heat the 32 kHz crystal pads to melt the solder that will get rid of water. The crystal is the metallic looking device near the microcontroller close to the PC edge.

Question: To confirm while in my house I had no problem with the SBMS0 booting up, after being installed in my camper for a couple of days when I finally installed the battery tonight I got the same screen with one line and no bootup. I live in Corpus Christi Texas on the coast and it is very humid here. I put a heat gun a couple of feet away and made sure it was only making the area warm at the unit and after a couple of hours, it booted up and is now up and running. It appears as though the unit is extremely sensitive to humidity and having it outside in the camper was enough for it to not boot. I am assuming that once up and running it will no longer be an issue.

Answer: The SBMS (all models) are designed for indoor use and non-condensing humidity. As soon as water condenses on the PCB that will influence the functionality. It seems like the real time clock is the most sensitive but other stuff can be also influenced if water droplets form inside.

While running, the self-consumption of the SBMS will usually keep the inside of the SBMS0 dry plus the crystal once it starts to oscillate it will be harder to be affected. Still, very high humidity to the point of condensing is not great for electronics and corrosion may be a problem also over long period.

10 Resources

10.1 Electrodacus-Specific

The Electrodacus Website: https://electrodacus.com

Youtube:

Electrodacus -Open Source- Solar Battery Management System: Introduction and Setup

Forums and discussion groups

diysolarforum > Solar BMS thread (SBMS0 SBMS40 SBMS120 and so on)

https://groups.google.com/electrodacus

Other:

https://github.com/bdlow/sbms0 - a repository of explanatory information (that has already been incorporated into this document).

https://github.com/armageddon421/electrodacus-esp32 - wifi module repository

10.2 Great LifePO4 Resources

Great, knowledgable content:

http://nordkyndesign.com/category/marine-engineering/electrical/lithium-battery-systems/

https://marinehowto.com/