

WAKESPEED[®] OFFSHORE



Communications and Configuration Guide

TABLE OF CONTENTS

About this guide	3
ASCII communications	4
CAN (Control Area Network) Communications	7
Receiving data FROM the regulator:	9
Sending data TO the regulator:	21
Appendix A: CAN enabled BMS	52
Appendix B: CAN messages	56
Appendix C: Details of CPE (Charge Profile Entries)	64
Appendix D: Default System Configuration	68
Appendix E: Error codes and meaning	70

ABOUT THIS GUIDE

The Wakespeed® Offshore series of product offerings features a common set of configuration and communications capability. This guide is used to document both Serial (ASCII) communications and CAN (Control Area Network) capabilities. Refer to the individual devices user's guide for additional details. Using the information in this guide one can access advanced configuration capabilities of the WS500 Alternator Regulator, as well as create a vertical stacked solution with tight integration between the regulator, BMS devices, Displays and engines. More so, this guide will be helpful in the development of supporting applications to aid in the configuration and use of the WS500 Alternator Regulator.

A note of NMEA2000™

NMEA2000 is a registered trademark of National Marine Electronics Association. In this guide there are references to NMEA2000, which are an artifact of the underlying J1939 support library used by the WS500 Alternator Regulator. The WS500 Alternator Regulator makes no claim of the correctness, compatibility, nor support for any NMEA2000 installation.

A note of edits and revisions

As this document is revised, new additions or capability will be placed in **RED** text. It is hoped this will aid in the quick assessment of new features. This release is intended for use with Firmware v2.2.0 or above.

Notable changes in 2.2.0:

- Post-Float will now restart a new charge cycle (move to Bulk) upon timeout of timer limit vs. moving to Float mode.
- Default CPE #8 (LiFeP04) modified to better match most commonly requested charge profiles
- Default Alternator Temperature raised from 90c to 100c
- Greatly expanded CAN based BMS integration, including additional protocols as well as optional BMS aggregation.
- Recording of last hard fault (See new \$RLF: command)

ASCII COMMUNICATIONS

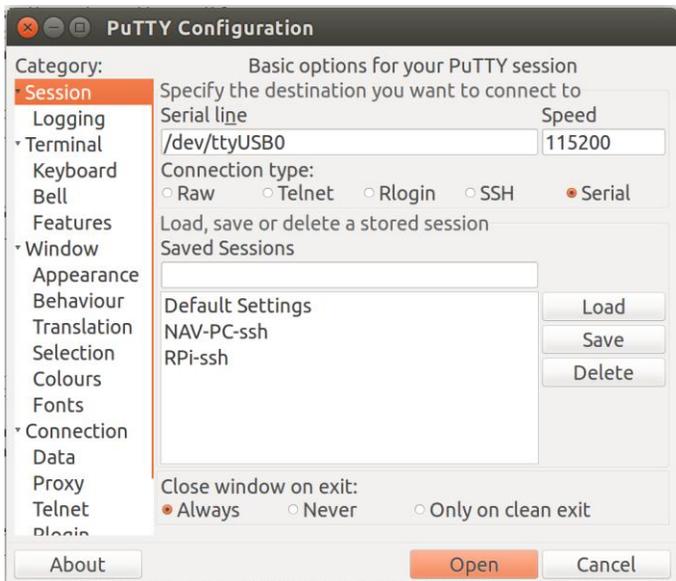
The WS500 Alternator Regulator features a Micro USB port to allow for advanced monitoring / diagnostics, configuration, as well as firmware updates. Simply connect a USB cable (making sure it is a proper data USB cable, and not a charger-only cable) between your computer and the USB micro port. Any needed drivers should install automatically, though you may need a connection to the internet depending on your operating systems.

Terminal Programs

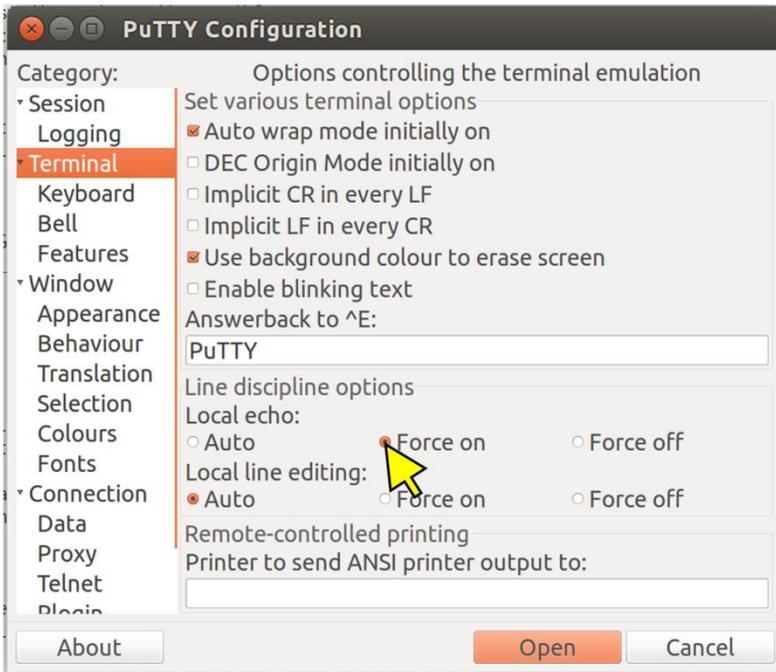
Many operating systems have built in terminal programs. After connecting the regulator to the computer you can use one of these to open a communications window to the WS500 Alternator Regulator. A special note, some terminal programs do not send a complete end-of-line terminator (CR+LF). To support these environments, the WS500 Alternator Regulator will recognize the character '@' as an alternative EOL indicator.

Remember that any configuration changes you make to the regulator may not take effect until the regulator is restarted. When you have finished, make sure to issue the \$RBT: command to not only assure changes are saved to the regulator's non-volatile memory, but that the changes are then utilized by the regulator. (refer to \$RBT: - ReBooT system on page 49). After rebooting the WS500 Alternator Regulator verify the changes you sent were recognized by the regulator by inspecting the various status strings.

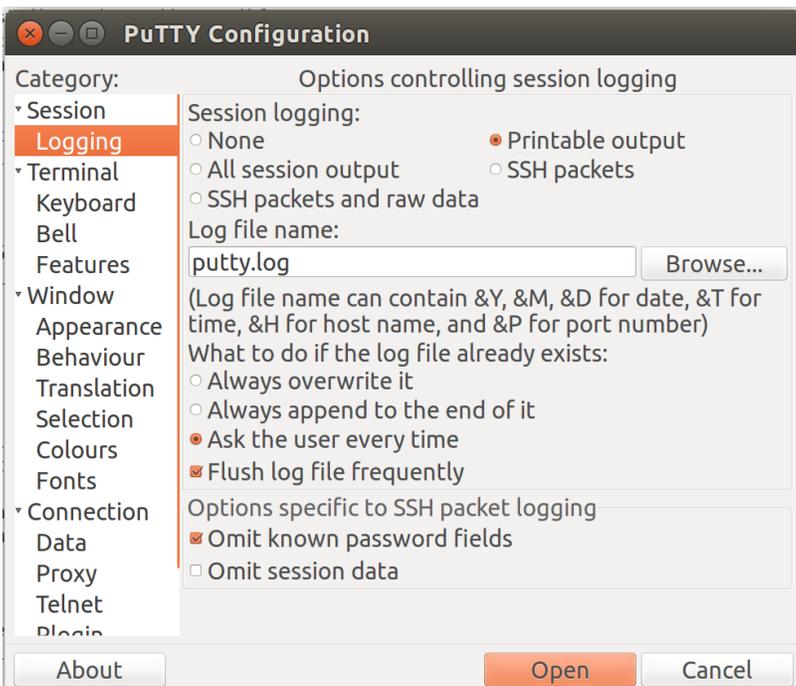
Putty: A versatile option is the free 'Putty' program (www.putty.org). It supports a wide range of OSs and includes a very nice logging function. To use connect up the USB cable and start Putty. Configure as shown here – selecting the Serial Line which your USB serial port is associated with, and setting the speed to 115200 and clicking the Serial Connection Type radial button.



Next click on the Terminal category and click the Forced-on button for local echo (as indicated on the following page), this will assure you can see what you are typing.



Finally, if you wish to keep a logfile of the session click on the Logging category, enter a file name and select the 'Printable Output' button as shown here:



Logging sessions is very helpful for debugging your installations, the files are comma separated and easily import into Excel using the import wizard specifying commas (,) as the separator. Refer to *Receiving data FROM the regulator:* for details on the output.

Once you have done your configuration press the Open button to start the terminal session.

Bench-top Configuration: When a USB cable is connected to the WS500 Alternator Regulator power is supplied to the logic portion of the hardware. This allows you to do bench-top configuration before completing the installation on the actual alternator. Make sure to do a \$RBT: command as your last step. After the regulator reboots make sure to verify your changes before installing the regulator in a live installation.

CAN (CONTROL AREA NETWORK) COMMUNICATIONS

The WS500 Alternator Regulator features CAN (Control Area Network) ports. Developed in the 1980's by Bosch and targeted towards the transportation sector, CAN is now one of the most widely deployed communications standards covering not only the Transportation sector but also used in Industrial, Heavy Industry, Farming, Medical, Consumer, and more. Over a billion CAN nodes have been deployed, with modern automobiles contained upwards of 100+ individual nodes each! It is a proven reliable and robust communications standard with many features to assure deterministic and prioritized communication resulting in a solid and proven reliable communications backbone.

Utilizing CAN devices are able to integrate into a 'System' where each works in cooperation with the others. Further, the CAN allow simple and reliable way to connect computers or displays for ongoing monitoring and easy configuration. The WS500 Alternator Regulator utilizes standards covering physical wiring, message content and other communications standards. These include:

- CAN Specification 2.0b / ISO-11898
- CiA 303
- SAE J1939
- OSEnergy (Open Systems Energy - derived from the RV-C standard)

OSEnergy (Open Systems Energy) is an architectural specification whose aim is to provide a framework for the design, deployment, and operation of charging sources associated with a DC battery. Allowing them to work together in a 'systems' approach while meeting the full requirements of an associated battery as well as concurrently supplying house power needs in a consistent and efficient way. You can learn more here: <https://github.com/OSEnergy/OSEnergy>

Through the application of these standards the WS500 Alternator Regulator is able to deliver several key benefits, including:

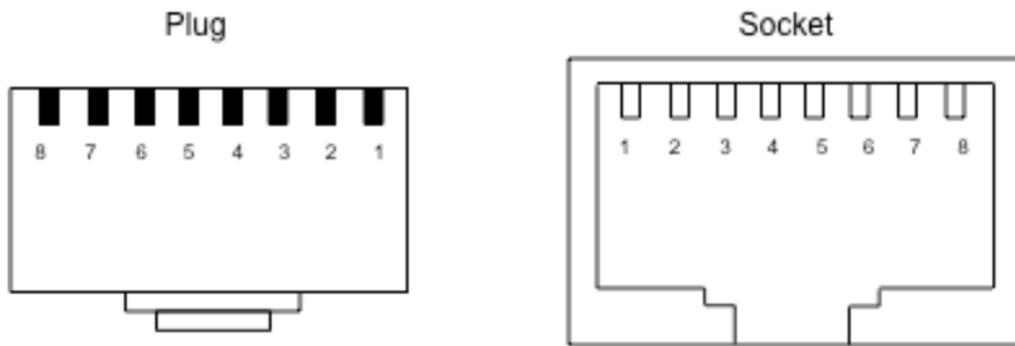
- Coordination of charging goals and objectives; all devices work towards the SAME goal vs. fighting each other.
- Tight BMS integration. CAN communications allows for the WS500 Alternator Regulator to fully integrate with the needs and directions of a BMS at levels unattainable using simple 'Charge Enable' wires.
- Prioritization of charging sources, e.g.: Utilization of Solar to its maximum capability while filling in the remaining energy needs from an engine driven alternator - thereby saving fuel.
- Remote sensing / Port Expander: The WS500 Alternator Regulator is able to take advantage of the CAN communications capability to transfer real-time battery status: voltage, amperage, and temperature as well as operational status (e.g., off-line in the case of a LiFePO4). By using this capability wiring and installations may be simplified.
- Self healing / fail over: Ability to self-recover from a failed, removed, or turned off device. The system continuously monitors all devices and adjusts as needed.
- 'Get-Home' total system failure mode: In the event of a catastrophic total system communications failure, the WS500 Alternator Regulator will fail-to-safe and operate in a stand-alone mode. Allowing for continued charging, but perhaps with less optimization and longer times needed.

CAN wiring

Use good quality CAT-5, CAT-5e, or CAT-6 cable to connect between devices in a daisy-chained fashion plugging into one of the two RJ45 connectors on the regulator. At *each end* of the daisy-chain install a terminator plug into the open RJ45 connector. It is important that the CAN bus be a single end-to-end chain with termination at each end. DO NOT connect an extra CAT-5 cable between the end devices making a loop – instead make sure each end point has one open RJ45 connector and then plug in the terminators.

The total length of the CAT-5 daisy-chain should be kept under 100M (300') with no more than 100x nodes total for best reliability.

The RJ45 connectors follow the CiA-303 standard, as shown here:



RJ45 connector

Pinning for RJ45 connector

Pin	Signal	Description
1	CAN_H	CAN_H bus line (dominant high)
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_GND	Ground / 0 V / V-
4	(SFTY_STOP)	Optional - Machine Enable / Emeryg Stop signal (Extension of CiA-303 spec)
5	-	Reserved
6	(CAN_SHLD)	Optional CAN Shield
7	(GND)	Optional ground
8	(CAN_V+)	Optional CAN external positive supply (dedicated for supply of transceiver and optocouplers, if galvanic isolation of the bus node applies) NOTE For recommended range of external power supply see clause 5.4

Figure 1 – CiA 303 CAN RJ45 connector specification

Connect the CAN_H and CAN_L signals. If supported, CAN_SHLD may also be optional connected as needed. It is generally NOT recommended to connect the CAN_GND to anything, as this may create a ground loop between it and ALT-

RECEIVING DATA FROM THE REGULATOR:

Receiving data FROM the regulator:

		9
AST; -- ALTERNATOR STATUS		10
CST; -- CAN STATUS		12
CPE; -- CHARGE PROFILE ENTRY		14
NPC; -- NAME & PASSWORD CONFIGURATION		16
SCV; -- SYSTEM CONFIGURATION		17
SST; -- SYSTEM STATUS		19
FLT; -- FAULTED		20
AOK; -- ACKNOWLEDGE		20
DBG; -- DEBUG		20
RST; -- RESET		20

All status outputs are suspended during the receiving and processing of a command string. In this way, a command which expects a response (ala \$RSC;) can be assured the next string sent back by the regulator is the response to the requesting command (though one should still do error checking and validation, as the simple regulator will often ignore commands containing a syntax error in them)

Formats are in clear ASCII using comma separated fields. Note the presence of double commas (separated by a space) between major 'sections', this is to simplify manual reading of the strings. Each string is delivered as one continuous line with a CR/LF termination.

Additional details of each status may be discovered by examine the command string for changing those parameters.

AST; -- ALTERNATOR STATUS



AST: “AST;, Hours, , BatVolts, AltAmps, BatAmps, SystemWatts, ,TargetVolts, TargetAmps, TargetWatts, AltState, ,BTemp, ATemp, ,RPMs, , AltVolts, FTemp, FAmps, FLD%”

Hours: Time regulator has been powered up, in hours and fraction (to 2 digits) of hours.

BatVolts: Derived Battery Volts, in volts and fractions of volts (to 1mV resolution). Used to decide charge mode changes.

AltAmps: Measured Alternator Amps, in Amps and fraction of Amps (to 1/10th of an Amp).

BatAmps: Derived Battery Amps being used to decide charge modes.

Voltage and current readings made by the WS500 Alternator Regulator are directly reported as *AltVolts* and *AltAmps*. Unless overridden by an external source (example via a Remote Battery Sensor) those same values will be assumed to be *BatVolts* and *BatAmps* and used by the regulator to make charge state decisions.

SystemWatts: Current measured System Watts being delivered.

TargetVolts: Volts the regulator is attempting to bring the battery to. This value is the ACTUAL voltage value being driven to, and reflected the adjusted Charge profile entry and the sysVolts index value.

TargetAmps*: Amps the regulator will limit the alternator to. This value is the ACTUAL amperage being driven to, and reflecting the derating and half power mode adjustments.

TargetWatts*: Watts the regulator is actually working to limit the system to.

AltState: Current state of the Alternator, per the following table:

Value	
0,1,4	- Alternator Off
2,3	- Alternator FAULTED (See Fault Code)
5	- In special DC-Disconnected CV mode.
9	- In CONFIGURATION mode
10	- Alternator Standby mode, or in delay mode while engine warms up.
11,15	- Ramping towards BULK mode.
12,20	- In BULK mode
21	- In ACCEPTANCE mode
22	- In OVER CHARGE mode
30	- In FLOAT mode
31	- In FORCED_FLOAT mode (via Feature_in pin and CPE = #8)
36	- In OFF (Post Float) mode
38	- In EQUALIZE mode
39	- In CVCC mode (only available in system under direction of CAN master)

- BTemp: Measured temperature of NTC sensor attached to B-port in degrees C or battery temperature received via external CAN sensor. -99 indicates temperature has not been measured, NTC sender has failed, not attached, and there is no remote temperature information available via the CAN connection.
- ATemp: Measured temperature of NTC sensor attached to A-port in degrees C. -99 indicate temperature has not been measured, or NTC sender has failed. -100 indicates the Alternator temp NTC probe is shorted (to select ½ power mode)
- RPMs: Measured RPMs of engine (Derived from Alternator RPMs and the Engine/Alternator drive ratio)
- (The following additions are available with Firmware version 1.0.0 and above)
- AltVolts: Measured Alternator Volts, in volts and fractions of volts (to 1mV resolution)
- FTemp: If equipped, this is the temperate of the FETs in degrees C.
-99 indicated FET temperate cannot be measured.
- FAmps: If equipped, this is a measurement of the current (amperage) being delivered to the field.
-99 indicated field current is not being measured.
- FLD %: % (0..100%) field is being driven.

*Note: * If the WS500 Alternator Regulator is configured with no limits for Alternator Amps and/or System Watts a self-impose limits of 1,000A / 15,000W as max values. AST; will report these working values.*

CST; -- CAN STATUS



“CST;, BatteryID, IDOverride, Instance, Priority, ,Enable NMEA2000?, Enable OSE?, ,AllowRBM?,IsRBM, ShuntAtBat?, ,RBM ID, IgnoringRBM?,**Enable_ALT_CAN**, ,CAN_ID, ,EngineID, BitRate, **Aggregate BMS**”

BatteryID: Battery number (or Instance) the regulator is associated with. 1..100

The following ‘convention’ is suggested – but not required:

1. Main House Battery
2. Primary Engine Starter battery (port engine)
3. Secondary House Battery
4. Secondary Engine Starter battery (starboard engine)
5. Generator Starter Battery
6. Forward Thruster battery
7. Aft Thruster Battery

IDOverride: Battery number (or Instance) is set via the DIP switches, however it is possible to ‘override’ the DIP switches using the \$CCN: command. (0= no override)

Instance: Charger Instance (1..13). Set with \$CCN: command (Default = 1)

Priority: Device priority, used to decide which devices should provide charging current, as well as who will be potential ‘master’ device. Set with \$CCN: command (Default = 70)

Enable NMEA2000?: 0 or 1, Is regulator configured to send NMEA-2000 like messages? (1 = Yes)

Enable OSE?: 0 or 1, Is regulator configured to send OSEnergy type messages? (1 = Yes)

By using the \$CCN: command, the user may disable portions of the CAN message stack. One would do this in cases where conflicts exist with existing devices on a shared CAN bus. An example might be the regulator is installed into an existing NMEA-2000 system, and it is desired to have NMEA-2000-like status be sent out; however some of the OSE messages cause issues with existing NMEA-2000 instruments. In this case the user may choose to disable OSE messages.

CAUTION: If OSE messages are disabled, all CAN-based value-add capabilities of the regulator will also be disabled. Including remote instrumentation, common charging goal, and charging device prioritization. **DISABLE OSEnergy MESSAGING WITH CAREFUL CONSIDERATION** and perhaps consider setting up isolated networks instead with a CAN bridge to forward the NEMA2000 messages to the proper NMEA2000 bus.

AllowRBM?: 0 or 1, Is regulator configured to attempt to act as the Remote Battery Master? (1 = Yes)

IsRBM?: 0 or 1, Does regulator currently think it is the Remote Battery Master? (1 = Yes)

ShuntAtBat?: 0 or 1, Does regulator currently think its shunt is directly connected to the battery? (1 = Yes, default = 0)

The WS500 Alternator Regulator is able to assume the role of the Remote Battery Master, thereby acting as the central coordinator for all charging sources. In practice, using the regulator as the RBM typically would occur only with small installations, twin engines installations are a common example. However, one is also able to configure a more extensive system where the WS500 Alternator Regulator is configured as a backup device. Set this via the \$CCN: command

RBM ID: Remote Battery Master ID: ID number of remote device which is currently recognized as the Remote Battery Master. 0 = WS500 Alternator Regulator has not associated itself with RBM.

IgnoringRBM?: 0 or 1, Is the regulator ignoring the Remote Battery Master? (1 = Yes)

If the Remote Battery Master sends information which seems unbelievable, this flag will be set and the regulator will ignore it. **Such a condition indicates something is wrong in the overall system and that should be investigated and resolved.** Conditions which will cause this fault include:

- Indicated Battery Voltage too high, or too low. (8..18v for normalized 12v battery)
- Indicated Battery Current too high (> +/- 2,000A)
- Voltage difference between battery and alternators > 1.5v (indicating issue with alternator wiring)
-

Enable_ALT_CAN: Bit-mask enabling alternative CAN based remote sensing and/or RBMs. See \$CCN: on page 45 for details.

CAN_ID: This is the current CAN Node ID, or node address which the WS500 Alternator Regulator has been assigned.

EngineID: Engine ID (or number) the WS500 Alternator Regulator with the engine it is mounted on. Used for monitoring J1939 RPMs messages as well as sending NMEA2000 RPMs back out. Note by convention EngineID = 0 is the default value.

BitRate Data communication rate CAN is set for.

Aggregate BMS: Indicates optional aggregation of BMS / RBM's.

CPE; -- CHARGE PROFILE ENTRY



In response to RCP: command, this displays the current values of a Charge Profile Entry. Special note on Charge Profile Entries: All voltage and current values in Charge Profile tables are displayed in their normalized '12V' values. See Defining Charging Voltages and Amps for additional information.

```
"CPE; n, acptVBAT, acptTIME, acptEXIT, res1, , ocAMPS, ocTIME, ocVBAT, ocaEXIT, , floatVBAT, floatAMPS, floatTIME, floatRESUMEA, floatRESUMEAH, floatRESUMEV, , pfTIME, pfRESUME, pfRESUMEAH, , equalVBAT, equalAMPS, equalTIME, equalEXIT, , BatComp, CompMin, MinCharge, MaxCharge, , RdcVolts, RdcLowTemp, RdcHighTemp, RdcAmps, , floatSOC, , MaxAmps, , pfVBAT,"
```

n:	Charge Profile 'n' is being displayed/returned (1..8)
acptVBAT:	Target battery voltage during BULK and ACCEPT phase
acptTIME:	Time limit to stay in ACCEPT mode – in Minutes.
acptEXIT:	Amp limit to trigger exiting ACCEPT mode
res1:	Reserved for future use (dV/dT exit criteria, currently = 0, disabled)
ocAMPS:	Max Amps which will be supplied by during OVERCHARGE mode.
ocTIME:	Time limit to stay in OVERCHARGE mode – in Minutes.
ocVBAT:	Target battery voltage during OVERCHARGE phase
ocaEXIT:	Amp limit to trigger exiting OVERCHARGE mode
floatVBAT:	Target battery voltage during FLOAT phase
floatAMPS:	Max Amps which will be supplied by during FLOAT mode.
floatTIME:	Time limit to stay in FLOAT mode – in Minutes.
floatRESUMEA:	Amp limit to trigger resumption of BULK charge mode
floatRESUMEAH:	Amp Hours withdrawn after entering Float to trigger resumption of BULK charge mode
floatRESUMEV:	Volt limit to trigger resumption of BULK charge mode

Note: If the regulator is in FORCED_FLOAT mode via the FEATURE-IN pin, then none of the above checks to exit float mode (e.g., floatTIME) will be performed. However, regulation will still occur to *floatVBAT* and *floatAMPS*.

pfTIME: Time limit to stay in POSTFLOAT mode – in Minutes, before resuming FLOAT charge mode.

pfRESUME: Battery Voltage that will trigger resumption of FLOAT charge mode

pfRESUMEAH: Amp Hours withdrawn after entering Post Float to trigger resumption directly to BULK charge mode

equalVBAT: Target battery voltage during EQUALIZE phase

equalAMPS: Current limit of Alternator while in EQUALIZE mode

equalTIME: Time limit to stay in EQUALIZE mode – in Minutes.

equalEXIT: Amp limit to trigger exiting EQUALIZE mode

BatComp: Temperature Compensations value per 1-degree C (normalized to '12v' battery)

CompMin: Minimum temperate to apply compensation at. In degrees C

MinCharge: Minimum temperate to charge the battery at, below this will force into FLOAT mode.

MaxCharge: Maximum temperate to charge the battery at, above this will force into FLOAT mode.

RdcVolts: Battery low Volts trigger for Reduced Charging

RdcLowTemp: Battery low Temperature trigger for Reduced Charging.

RdcHighTemp: Battery high Temperature trigger for Reduced Charging.

RdcAmps: Current limit while in Reduced Charging state.

flaotSOC: SOC level to trigger resumption of BULK charge mode .

MaxAmps: Maximum battery charge current during any charge phase.

pfVBAT: Target battery voltage during POST-FLOAT phase

NPC; -- NAME & PASSWORD CONFIGURATION



"NPC;, Use BT?, Name, Password, , DeviceID"

use BT?	0 or 1: Enable Bluetooth? (1 = yes)
Name:	Name of Regulator (Used for CAN device ID) (ASCII up to 18 characters)
Password:	Password (ASCII up to 18 characters)
DeviceID	Semi-Unique device ID of WS500 Alternator Regulator

SCV; -- SYSTEM CONFIGURATION



"SCV;, Lockout, reserved, RevAmp, SvOvr, BcOvr, CpOvr, ,AltTempSet, drtNORM, drtSMALL, drtHALF, PBF, ,Amp Limit, Watt Limit, , Alt Poles, Drive Ratio, Shunt Ratio, ,IdleRPM, TachMinField, Warmup Delay, Required Sensor, DC_Disconnected_VBat"

Lockout: Current lockout level. (0..2), see \$SCO: command.

Reserved: Always = 0;

RevAmp: 0 or 1: Reverse polarity of Amp Shunt readings? (1 = yes)

SvOvr: System Voltage auto-detect (=0), or force (1.0x .. 4.0x → 12v .. 48v)

BcOvr: Override Battery Capacity DIP switches (Dip 5/6). (0.00 = No)

CpOvr: Override Charge Profile DIP Switches (Dip 2..4) (0 = No)

AltTempSet: Target max running setpoint for Alternator, in degrees C

drtNORM: Normal Amp reduction (de-rating) fraction

drtSMALL: Amp reduction (de-rating) fraction when in SMALL - MODE

drtHALF: Amp reduction (de-rating) fraction when in half-power mode.

PBF: Pull-back factor, for reducing Field Drive at lower RPMs.

Amp Limit: Defined Alternator size, or -1 to enable auto-sizing. Set this = 0 for installations where Alternator Sizing is not to be regulated (ala, battery focused installations).
Note: During startup, and unless defined, this value will present: 1,000

Watt Limit: Defined System size, or -1 to enable auto-sizing. Set this = 0 for installations where Watts loading is not to be regulated (ala, battery focused installations).
Note: During startup, and unless defined, this value will present: 15,000

Alt Poles: Number of poles on Alternator

Drive Ratio: Ratio of engine and alternator drive pulley

Shunt Ratio: Amp Shunt ratio in Amps / mV

IdleRPM: Idle RPM value used as basis for Field Drive Reduction at lower RPMs.

TachMinField: Minimum % of field drive that will be applied if TACH MODE is enabled.

Warmup Delay: Number of seconds after power on before entering RAMP mode.

Required Sensor: Key indicating critical sensors which have been configured via the \$SCA: command

DC_Disconnected_VBat: Indicated how charging device should behave when it received a DC-Disconnect message via the CAN.

SST; -- SYSTEM STATUS



“SST;, Version , ,Small Alt Mode?, Tach Mode?, , CP index, BC Mult, SysVolts, ,AltCap, CapRPMs, , Ahs, Whs, ,ForcedTM?”

Version: Firmware revision identifier. Will have format of “AREG” followed w/o a space by the version number. E.g., “AREG1.0.1”

Small Alt?: 0 or 1, has the user selected Small Alternator Mode? (1 = yes)

Tach Mode?: 0 or 1, has user selected Tach Mode? (1 = yes)

CP Index: Which Charge profile (1..8) is currently being used?

BC Mult: What adjustment factor for Battery Amp Hour Capacity (1-10x) is currently being used? Fractional values may also be used to fine tune the system to a given battery size. This needs to be entered via the \$SCO: command.

SysVolt: Detected system voltage. Adjusts target Charge Profile Volts per the following table:

SysVolt	Detected System Voltage	Charge Profile VOLTAGE Adjustment Factor
1	12v	1x
2	24v	2x
2.67	32v	2.667x
4	48v	4x

Fractional values may also be used to support battery voltages such as 8v, 32v, 42v. Those values will need to be selected manually via the \$SCO: command.

Alt Cap: If regulator is configured to auto-determine the capacity of the alternator, this will be the current high-water mark noted.

CapRPMs: And this will be the RPMs at which that capacity was noted at.

AHs: The number of Amp-Hours that have been produced in the current charge cycle.

WHs: The number of Watt-Hours produced in the current charge cycle.

ForcedTM?: 0 or 1, has user forced Tach-Mode on via the \$SCT: command.

FLT; -- FAULTED

FLT: "FLT;, FaultCode, RequiredSensorStatus "

System has Faulted, fault code number.

RequiredSensorStatus is a combined number following the pattern as defined by the \$SCA command and detailed in 'Table 1 - Required Sensor Encoding ' on page number 36

Following this, the AST, SST and SCV will be printed, as well as the currently active CPE.

AOK; -- ACKNOWLEDGE

Sent after a successfully received change command (\$CPx, or \$SCx), \$MSW, or \$EDB

DBG; -- DEBUG

Special string with extra internal parameters for use in debugging internal operations.

RST; -- RESET

Regulator has been requested to reset. This can take up to 10 seconds to complete.

SENDING DATA TO THE REGULATOR:

All commands begin with the character '\$', contain 3 letters(CAPS) followed by a ':' and then parameters as requested by the command. All must end with a CR (or CR/LF) or may optional be terminated with the character '@'. A complete 'string' must be received within 60 seconds from the '\$' to the ending '@'/CR/LF, else the regulator will abort the capture of that string and begin looking for a new '\$' starting character.

Sending data TO the regulator:	21
\$CPA:n - Change ACCEPT parameters in CPE user entry n (n = 7 or 8)	 23
\$CPO:n - Change OVERCHARGE parameters in CPE user entry n (n = 7 or 8)	 24
\$CPF:n - Change FLOAT parameters in CPE user entry n (n = 7 or 8)	 25
\$CPP:n - Change POST-FLOAT parameters in CPE user entry n (n = 7 or 8)	 27
\$CPE:n - Change EQUALIZE parameters in CPE user entry n (n = 7 or 8)	 28
\$CPB:n - Change BATTERY parameters in CPE user entry n (n = 7 or 8)	 29
\$CPR:n - RESTORES Charge Profile 'n' to default values	 30
\$RAS: - Request All Status back	 30
\$RCP:n -Request to send back CPE entry #N (n=1..8) **	 31
\$RCP:0 - Request to send back current selected CPE**	 31
\$RLF: - Request Last Fault	 32
\$SCA: - Changes ALTERNATOR parameters in System Configuration table	 33
\$SCT: - Changes TACHOMETER parameters in System Configuration table	 39
\$SCO: - Override features	 41
\$SCN: - Changes NAME (and PASSWORD)	 43
\$SCR: - RESTORES System Configuration table to default	 44
\$CCN: - Change parameters in the CAN Configuration table	 45
\$CCR: - RESTORES CAN Configuration table to default	 48
\$MSR: - RESTORE all parameters (to as defined at program compile time).	 48

\$EDB: - Enable DeBug serial strings



\$RBT: - ReBooT system



\$FRM: - Force Regulator Mode



Defining Charging Voltages and Amps – All volts and amps used in the Charge Profile Entries are represented by a normalized 12v 500Ah battery and are automatically scaled depending on the sampled battery voltage at startup and the setting of the Battery Capacity DIP switches.

Take care that the MAXIMUM length of a string is fixed at 70 characters, including the line termination character(s). When assembling a command to send make sure not to exceed this length, remove any extra spaces if present to assure total length is under 70 characters.

****Note on Requesting Status commands.** All 'Request' commands will reply via the serial port. In addition, IF the request arrived via the J1939 CAN 'Terminal' DGN (17E00h), a copy of the reply will also be returned to the requesting CAN node. This is useful to gain access to the advanced ASCII setup parameters of the WS500 Alternator Regulator via the CAN.

\$CPA:n - Change ACCEPT parameters in CPE user entry n (n = 7 or 8)

This command will cause the ACCEPT (and BULK) portion of a Charge Profile Entry to be updated. Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

`$CPA:n <VBat Set Point>, <Exit Duration>, <Exit Amps>, <Reserved>`

n: (7 → 8) ‘n’ is the Charge Profile Table Entry that will be modified. Use range 7 to 8.

VBat Set Point: <FLOATING POINT NUMBER (0.0 → 20.0) > Voltage the Regulator will use during ACCEPT phase. When this voltage has been reached, the regulator will transition from BULK to ACCEPT phase. This value is a floating-point number and entered for a normalized 12v system (See note: Defining Charging Voltages and Amps)

Exit Duration: <WHOLE NUMBER (0 → 600 (10 hours)) > After entering ACCEPT phase, a timer will be started. After ‘ExitDuration’ minutes have expired ACCEPT mode will exit and the regulator will move to OVER-CHARGE mode. Setting ‘ExitDuration’ = 0 will disable time based exiting of ACCEPT mode and only AMP based monitoring will be used.

Exit Amps: <WHOLE NUMBER (-1 → 200)> After entering ACCEPT phase, delivered amps will be monitored and if they fall to (or below) ‘ExitAmps’ ACCEPT mode will exit and the regulator will move to OVER-CHARGE mode. This is providing that the battery voltage is at the target VBat Set Point above (to prevent early exiting from low amps being delivered as a result of the engine slowing down to say very slow idle).

Setting ‘ExitAmps’ = 0 will disable amp-based exiting of ACCEPT mode and only time monitoring to ‘ExitDuration’ will be used.

Setting ‘ExitAmps’ = -1 will disable amp-based exiting of ACCEPT mode and time monitoring of ‘ExitDuration’ will be used as above. In addition, when the time spend in Acceptance mode has exceed 5x the duration spent in Bulk mode, the regulator will also trigger an exit. (Adaptive Acceptance).

Note: If you set BOTH ‘ExitDuration’ & ‘ExitAmps’ = 0, the regulator will bypass ACCEPT mode.

Reserved: <0> Must be 0.

Example:

```
$CPA:7 14.5, 200, 40, 0@      #7: 14.5VOLTS, EXIT AFTER 200 MINUTES OR UNDER 40AMPS
$CPA:8 12.4, 0, 20, 0@      #8: 12.4 VOLTS, EXIT ONLY ON AMPS UNDER 20
$CPA:810.4,0,20,0@        #8: 10.4 VOLTS, EXIT ONLY ON AMPS UNDER 20.
      (SHOWN WITH OPTIONAL '@' FOR ARDUINO IDE TERMINAL SUPPORT)
```

\$CPO:n - Change OVERCHARGE parameters in CPE user entry n (n = 7 or 8)

This command (with its parameters) will cause the OVER-CHARGE (Sometimes referred to as FINISH phase) portion of a Charge Profile Entry to be updated. Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

```
$CPO:n <Limit Amps>, <Exit Duration>, <Exit VBat>, <Exit Amps >
```

n: (7 → 8) ‘n’ is the Charge Profile Table Entry that will be modified. Use range 7 to 8.

Limit Amps: <WHOLE NUMBER (0 → 50)> After entering OVER-CHARGE phase, delivered amps will be monitored and regulated to ‘LimitAmps’. Setting ‘LimitAmps’ = 0 will disable OVER-CHARGE mode.

Exit Duration: <WHOLE NUMBER (0 → 600 (10 hours)) > After entering OVER-CHARGE phase, a timer will be started. After ‘ExitDuration’ minutes have expired OVER-CHARGE mode will exit and the regulator will move to FLOAT mode. Setting ‘ExitDuration’ = 0 will disable OVER-CHARGE mode.

Exit VBat: <FLOATING POINT NUMBER (0.0 → 20.0) > Once battery voltage reached ‘ExitVBat’, OVERCHARGE phase will be exited. This value is a floating-point number and entered for a normalized 12v system (See: Defining Charging Voltages and Amps). Setting ‘ExitVBat’ = 0 will disable OVER-CHARGE mode.

Exit Amps: <WHOLE NUMBER (0 → 50)> If defined, ‘ExitAmps’ will keep the regulator in OVER-CHARGE phase until the battery acceptance current has fallen to (or below) this value. During this time voltage will remain at ‘ExitVBat’. ‘ExitAmps’ will NOT override the overall ‘ExitDuration’ transition criteria. Set = 0 to disable this option.

\$CPF:n - Change FLOAT parameters in CPE user entry n (n = 7 or 8)



This command (with its parameters) will cause the FLOAT portion of a Charge Profile Entry to be updated. Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

\$CPF:n <VBat Set Point>, <Limit Amps>, <Exit Duration>, <Revert Amps>, <Revert Amp-hours>, <Revert Volts>, <Revert SOC>

n: (7 → 8) ‘n’ is the Charge Profile Table Entry that will be modified. Use range 7 to 8.

VBat Set Point: <FLOATING POINT NUMBER (0.0 → 20.0)> Voltage the Regulator will use during FLOAT phase. This value is a floating-point number and entered for a normalized 12v system (See note: Defining Charging Voltages and Amps)

Limit Amps: <WHOLE NUMBER (-1 → 50)> While in FLOAT phase delivered Amps will be monitored and regulated to ‘LimitAmps’. Setting ‘LimitAmps’ = -1 will disable this feature and only ‘VBatSetPoint’ will be regulated.

Exit Duration: <WHOLE NUMBER (0 → 30000 (500 hours))> After entering FLOAT phase, a timer will be started. After ‘ExitDuration’ minutes have expired FLOAT mode will exit and the regulator will move to POST-FLOAT mode. Setting ‘ExitDuration’ = 0 will cause the regulator to remain in FLOAT mode unless triggered by another exit criteria.

Revert Amps: <WHOLE NUMBER (-300 → 0)> While in FLOAT mode, if ‘RevertAmps’ are exceeded it is an indication that a large load has been placed on the battery and current is being withdrawn, the regulator will re-start a charge cycle, looping back to BULK mode. Setting ‘RevertAmps’ = 0 will disable this feature.

RevertAmps is most useful in the case where the amp shunt is placed on the battery, as when the amp draw from the battery exceeds *RevertAmps*, it is a clear indication energy is being drawn from the battery. In this case, set *RevertAmps* equal to the number of amps being drawn from the battery that should be used to trigger a revert to bulk.

In cases where the amp shunt is installed on the alternator, this can also be of use by sizing *RevertAmps* to a value slightly above expected house load values. However, perhaps a better indication is to set this to =0, and use *RevertVolts*.

Revert Amps-hours: <WHOLE NUMBER (-250 → 0)> After entering FLOAT mode if the accumulated number of Amp Hours removed from the battery exceeded ‘RevertAmp-hours’ the regulator will re-start a charge cycle, looping back to BULK mode. Setting ‘RevertAmp-hours’ = 0 will disable this feature

This is another way to indicate the need to restart charging of the battery, and perhaps a better approach than raw *RevertAmps*, but it is only usable if the amp shunt is placed on the battery.

Revert Volts: <FLOATING NUMBER (0.0 → 20.0)> While in FLOAT mode, if battery voltage drops below '*RevertVolts*' we assume this indicates a large load has been placed on the system and the regulator will re-start a charge cycle, looping back to BULK mode. Setting '*RevertVolts*' = 0 will disable this feature.

Revert SOC: <WHOLE NUMBER (0 → 100)> While in FLOAT mode, if the batteries SOC (State Of Charge) as reported by a CAN attached BMS or other RBM drops below '*RevertSOC*' the regulator will re-start a charge cycle, looping back to BULK mode. Setting '*RevertSOC*' = 0 will disable this feature.

In determining to exit Float Mode, a rolling average value for measured Amps and Volts is used. This way short term events (e.g., a surge of a refrigerator starting up and before the Alternator can respond) will not pull the regulator out of Float mode. Note also that the revert Amp –hours are a negative value, and measure the number of AHs removed from the battery after entering Float mode.

\$CPP:n - Change POST-FLOAT parameters in CPE user entry n (n = 7 or 8)



This command (with its parameters) will cause the POST- FLOAT portion of a Charge Profile Entry to be updated. Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

\$CPP:n <Exit Duration>, <Revert VBat>, <Revert Amp-hours>, <VBat Set Point>

n: (7 → 8) ‘n’ is the Charge Profile Table Entry that will be modified. Use range 7 to 8.

Exit Duration: <WHOLE NUMBER (0 → 30000 (500 hours)) > After entering POST-FLOAT phase, a timer will be started. After ‘ExitDuration’ minutes have expired POST-FLOAT mode will exit and the regulator will revert to **FLOAT BULK** phase. Setting ‘ExitDuration’ = 0 will cause the regulator to remain in POST-FLOAT mode unless triggered by another exit criteria.

Revert VBat: <<FLOATING POINT NUMBER (0.0 → 20.0)> While in POST-FLOAT mode, if the system battery voltage drops below ‘RevertVBat’ it is an indication that a large load has been placed on the system and the regulator will re-start a charge cycle, looping back to BULK mode. Setting ‘RevertVBat’ = 0 will disable this feature.

Revert Amps-hours: <WHOLE NUMBER (-250 → 0)> After entering POST-FLOAT mode if the accumulated number of Amp Hours removed from the battery exceeded ‘RevertAmp-hours’ the regulator will re-start a charge cycle, looping back to BULK mode. Note that this trigger goes directly to Bulk, as opposed to back to Float mode. Setting ‘RevertAmp-hours’ = 0 will disable this feature

VBat Set Point: <FLOATING POINT NUMBER (0.0 → 20.0) > Voltage the Regulator will use during POST-FLOAT phase. This value is a floating-point number and entered for a normalized 12v system. If set = 0.0 (default), charging will be disabled during post-float mode.

\$CPE:n - Change EQUALIZE parameters in CPE user entry n (n = 7 or 8)



This command (with its parameters) will cause the EQUALIZATION portion of a Charge Profile Entry to be updated. Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

\$CPE:n <VBat Set Point>, <Max Amps >, <Exit Duration>, <Exit Amps>

n: (7 → 8) ‘n’ is the Charge Profile Table Entry that will be modified. Use range 7 to 8.

VBat Set Point: <FLOATING POINT NUMBER (0.0 → 25.0) > Voltage the Regulator will use during EQUALIZE mode. This value is a floating-point number and entered for a normalized 12v system (See note: Defining Charging Voltages and Amps). Setting ‘VBat’ = 0 will disable EQUALIZE mode.

MaxAmps: <WHOLE NUMBER (0 → 50)> Optional additional current limit while in EQUALIZE phase; the regulator will cap delivered AMPS to ‘MaxAmps’ . Setting ‘MaxAmps’ = 0 will disable this amperage capping.

Exit Duration: <WHOLE NUMBER (0 → 600 (10 hours)) > After starting an EQUALIZE phase, a timer will be started. After ‘ExitDuration’ minutes have expired EQUALIZE will emanate and the regulator will enter FLOAT mode. Setting ‘ExitDuration’ = 0 will disable EQUALIZE mode.

Exit Amps: <WHOLE NUMBER (0 → 50)> During EQUALIZE mode delivered Amps will be monitored and if it falls to (or below) ‘ExitAmps’ equalization will be terminated and the regulator will move to FLOAT mode. Note that as a precaution, Battery Voltage is not checked when sampling Equalization Exit Amps (as it is in Acceptance and Overcharge). It is up to the operator to keep the engine speed up and allow for a full equalization session to occur. Setting ‘ExitAmps’ = 0 will disable Amp based exiting of EQUALIZE mode and only Time monitoring will be used.

\$CPB:n - Change BATTERY parameters in CPE user entry n (n = 7 or 8)



This command (with its parameters) will cause the remaining portion of a Charge Profile Entry to be updated. Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

\$CPB:n <VBat Comp per 1°C>, <Min Comp Temp >, <Min Charge Temp>, <Max Charge Temp>, <Rdc Volts>, <Rdc Low Temp>, <Rdc High Temp>, <Rdc Amps>, <MaxBatAmps>

n: (7 → 8) ‘n’ is the Charge Profile Table Entry that will be modified. Use range 7 to 8.

VBat Comp: <FLOATING POINT NUMBER (0.0 → 0.1)> This is used to adjust all target VBat voltages based on the current Battery Temperature in 1 degree C increments. This value is a floating-point number and entered for a normalized 12v system at 25c. (See note: Defining Charging Voltages and Amps). Set = 0.0 to disable temperature based voltage compensation.

Min Com Temp: <WHOLE NUMBER (-40 → 40)> Additional compensation to battery target voltages will be stopped when the battery is at or below this temperature in degree C.

Min Charge Temp: <WHOLE NUMBER (-50 → 10) > **If the battery drops below this temperature, charging will be disabled. Once temperature rises above Min Charge Temp, a new charge cycle will start.**

Max Charge Temp: <WHOLE NUMBER (20 → 95) > If the battery reaches this temperature, the system will be **disabled to protect it. If the battery temperature continues to rise, the system will eventually FAULT when battery temperature exceeds Max Charge Temp by 20%. Note that Min and Max Temp limits will override any outside direction the WS500 regulator received; even if a CAN connected BMS is asking for charging, if either of these hard stop limits are exceed the regulator will enter Disabled mode.**

Rdc Volts: <FLOATING POINT NUMBER (0.0 → 12.0) > If the Battery Voltage is at or below this value, the regulator will work in Reduced Charge mode, where the maximum amps delivered to the battery will be limited to *Rdc Amps*. Set *Rdc Volts* = 0.0 to disable checking of battery voltage for Reduced Charging mode.

Rdc Low Temp : <WHOLE NUMBER (-99 → 20) > If the battery temperature falls to or below this value in degrees C, the regulator will enter Reduced Charge mode. Set = -99 to disable this test

Rdc High Temp : <WHOLE NUMBER (-99 → 95) > > If the battery temperature rises to or above this value in degrees C, the regulator will enter Reduced Charge mode. Set = -99 to disable this test

Rdc Amps : <WHOLE NUMBER (0 → 100) > When the regulator is in Reduced Charge mode, battery acceptance current will be limited to this value. Rdc Amps is defined relative to the normalized 500Ah battery and is adjusted by the system battery capacity. Set = 0 to disable Reduced Charge mode.

Max Bat Amps : <WHOLE NUMBER (0 → 2000) > If set, this will be the maximum amperage allowed to flow into the battery. Remember, this value is set relative to the 12v/500A ‘nominal’ battery and it

is also adjusted by the Battery Capacity Multiplier; so a value of 250 would cause the regulator to limit charging current to a 0.5c rate. Set = 0 (default) to disable Max Battery Amps

\$CPR:n - RESTORES Charge Profile 'n' to default values

Restores to default (values at compile time) Charge Profile Entry 'n'. After entry 'n' is restore, the regulator will be restarted automatically.

\$CPR:n

n: (7 → 8) 'n' is the Charge Profile Table Entry that will be restored. Use range 7 to 8.

\$RAS: - Request All Status back

This command will instruct the WS500 Alternator Regulator to send out via the Serial port a copy of the all known status strings. It is useful for external applications to capture the current status w/o needing to await the arrival of each string (which could take several minutes or more, depending on how status strings are spaced out).

\$RAS:

After all status strings have been sent, "AOK" will be sent to indicate the end.

\$RCP:n -Request to send back CPE entry #N (n=1..8) ** 

This command will instruct the WS500 Alternator Regulator to send out via the serial port the SAVED contents of the CPE entry N, where N is a number from 1 to 8. See “CPE: Charge Profile Entry” for description of resulting transmission.

\$RCP:n

\$RCP:0 - Request to send back current selected CPE** 

Special version of Request for CPE, this will send back the currently selected (via the DIP switched) CPE.

\$RCP:0

Note, the CPE entry sent back in response to a \$RCP: command will reflect the current values contained in FLASH memory which may not match what the regulator is currently working with. If a CPE has been modified and saved to FLASH, those modifications will be reflected. However, until the WS500 Alternator Regulator is rebooted it will not utilize those values. For current active targets being used, look at AST; and SCV; status strings.

Note on Change Requests to Charge Profiles: The only Charge Profile 7 or 8 (the two customizable entries) may be modified via an ASCII command. This is to reduce the potential for major errors in the regulator.

Also, take great care in setting these values, esp the exiting time and amp thresholds. Some of these thresholds can be disabled by setting to 0, disabling that threshold test. If both Amps & time values are disabled, it is possible for the regulator to stay in a full charge state indefinitely, likely causing damage to the battery. As the WS500 Alternator Regulator may be deployed with the amp shunt at the Alternator, or at the battery, some of the CPE entries will behave differently depending on which deployment model is used. And some entries might have no meaning. Great Flexibility results in Great Responsibility...

All Change Profile commands will reply with “AOK;” if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

\$RLF: - Request Last Fault



This command will instruct the WS500 Alternator Regulator to send a copy of the Last Known Fault information, including the Fault Number and a copy of the AST and CST strings at the time of the fault – preceded by two dots (..). After the last string is sent out, \$AOK will be sent.

If no recorded Fault exists, only the \$AOK will be sent. \$MSR will clear any saved last-fault information.

\$RLF:

After all status strings have been sent, "AOK" will be sent to indicate the end. Example of Low Voltage Fault:

```
$RLF:@  
  
..FLT;,14,0  
..AST;,0.00, ,0.06,-3.8,-3.8,0, ,13.10,100,15000,2, ,-99,-99, ,0, ,0.06,19,-99,0  
..CST;,1,0,2,70, ,1,1, ,1,0,1, ,0,0,0, ,129, ,0,0,0  
AOK;
```

\$SCA: - Changes ALTERNATOR parameters in System Configuration table



Used to update the system configuration table entries associated with the Alternator.

\$SCA: <reserved>, < Alt Target Temp >, <Alt Derate (norm) >,<Alt Derate (small) >,<Alt Derate (half) >, <PBF>, <Amp Cap >, <System Watt Cap. >, <Amp Shunt Ratio>, <Shunt Reversed?>,<Idle RPMs>,<Warmup Delay>,<RequiredSensors>

Reserved: <WHOLE NUMBER = 0> Set = 0

Alt Target Temp: <WHOLE NUMBER (15 → 150)> Operating temperature the regulator should attempt to keep the Alternator under in degrees C. If the Alternator temperature exceeds this value, the regulator will reduce field current to allow the alternator to return to a safe operating temperature. if the Alternator temperature continues to rise and exceeds this temperature by 10% the regulator will fault out and stop producing power.

Alt Derate(norm),

Alt Derate(small),

Alt Derate(half): <FLOATING POINT NUMBER (0.10 → 1.00) > These derating values are used to limit the alternator's maximum current output to some % (10% to 100%) of its demonstrated capability (see *Alt Amp Cap*). The three values correspond to the mode the Alternator:

- Normal - Condition when either of the other modes are not selected.
- Small Alternator Mode – selected via DIP switch 8 (or the override via \$SCT command)
- Half Power Mode – Selected by shorting the Alternator NTC temperature sensor wires.

In operation, De-rating values are applied to BOTH the Alt Amp Cap as well as the internal maximum field PWM drive. In this way, a smaller alternator is protected, even if the amp shunt is not connected.

PBF: < INTEGER (-1 → 10)> Pull-back factor for reducing Field Drive at lower RPMs. If the WS500 Alternator Regulator is able to determine RPMs (via the Stator wire), the Alternator Field Drive will be reduced when the regulator detects the engine is at Idle. At idle the max PWM will be capped at around 1/4 of full field, which should result in some current being produced. As RPMs are increased, this 'Field Drive Capping' will slowly be removed. PBF determines how quickly this pull-back is scaled off.

Set = 0 to disable this feature.

Set = -1 (DEFAULT) to cause Field Drive to be reduced to a maximum of 70% drive in the case where the WS500 Alternator Regulator is no longer able to measure RPMs via the Stator-in signal. This might be for example where an engine is operating at extremely low RPMs, below the cut-in point for the alternator. Or where the engine is no longer running. The 70% limit will only be enabled if at one time during operation the regulator was able to measure RPMs successfully.

For many engine / alternator combinations the default value of 1 should result in good operations. However, if you have installed a large alternator on a rather modest sized engine, you might notice the engine struggles when trying to increase RPMs from idle. In that case, increase the PBF value. A factor of 8x or so might be needed in the case of a small sail-boat engine with a large 150A or greater alternator (consider also using the Alt Amp Cap and/or System Watts Cap capabilities as well to restrict maximum engine loading at higher RPMs).

If the engine has a large capacity relative to the alternator size, consider reducing the PBF to 1. Doing so will allow a greater production of amps while at idle, while at the same time preventing the alternator from being driven at Full Field during low RPMs (and hence low cooling)

Finally, if your system matches an engine with great capability, and the alternator has good cooling / heat management – you can set the PBF factor = 0 to disable any capping of field drive while the engine is at idle. This will allow for maximum alternator output at idle, however if the regulator is enabled but the engine is not actually running, field drive will increase to Full Field until a fault check causes the regulator to reset. Do not leave the 'ignition' in the ON position, without the engine actually running to prevent this situation. It would be advisable to assure there is a temperature sensor attached to the alternator in this case – to prevent unintended overheating during prolonged idle periods.

Note: Field pull back is dependent upon the stator sensing wire being connected to the alternator. If the regulator is unable to reliably sense RPMs, all idle pull-back features will be disabled. Note also that one should make sure to configure the tachometer via the `$SCT:` command.

Amp Cap: <WHOLE NUMBER (-1→ 500) > With the Current Shunt located at the alternator (vs. the battery) The regulator will limit the Amperage output of the alternator to this value, after applying the '*Alt Derate xxx*' factors. There is no adjustment made to this value based on system voltage or selection of system battery size – the values declared will be used directly. A special feature is enabled by setting this = -1: the regulator will drive the alternator as hard as it can for a short period of time when 1st entering Bulk phase and in this way will auto-sample the alternator size based on its capabilities.

Do note with this option, there may be some interactions between the field % pullbacks and an attempt to reduce what is perceived as 'alternator current' pullback, for example during half-power mode.

With Amperage Cap disabled, reduced power modes will apply the scaling factor to the PWM duty cycle. It should be noted that there may not be a direct relation between reductions in PWM duty cycles and delivered Amperages – care should be used when setting up the system. (Default = 0, disabled)

System Watts Cap: <WHOLE NUMBER (-1 → 20000)> This regulator will limit the system wattage to this value. Its primary use is to protect the driving engine and/or belts – by limiting the maximum amount of Work the engine is asked to do in behalf of the alternator. (Work being a function of BOTH Volts and Amps, hence Watts). It may also be used to limit the total amount of power being delivered into the battery by all charging sources. There is no derating or adjustment made to this value based on system voltage or selection of system battery size. System Watts Capacity is used to after applying the 'Alt Derate xxx' factors. It is used to protect the alternator from over current usage. A special feature is enabled by setting this = -1, the regulator will drive the alternator as hard as it can for a short period of time when 1st entering Bulk phase. This will then be used to define the Amp Limit of the Alternator.

Note on Alt Amps and System Watts: You may set either of these parameters =-1 to allow the regulator to automatically calculate limits based on the sampled capability of the alternator, or set them = 0 to disable that feature. Though these two are interlaced, they are indeed separately monitored and adjustments to the Field PWM are made independently for each.

.(Default = 0, disabled)

Amp Shunt Ratio: <WHOLE NUMBER 500 → 20000)> Enter the ratio of your Amp measurement shunt in terms of AMPS / mVolts. e.g., if you have a 250A / 75mV shunt, you would enter 3333 (250/0.075). And you may adjust the number to allow for fine tuning of the Amp Shunt. e.g., if your shunt has a 3% error, you could enter 3433

Caution: Shunt Voltage is limited to +/-80mV. Do NOT exceed this value!

Shunt reversed?: <WHOLE NUMBER (0, or 1) > Allows software correction if the Amp Shunt was wired backwards. Set = 1 and Amp readings will have their polarity changed.

Idle RPMs: < INTEGER (0 → 1500) > Used in conjunction with PBF to manage Field Drive at lower RPMs. As RPMs rise above Idle RPMs, field drive will be increased at a rate determined by PBF. During normal operation, Idle RPMs can be detected automatically by the WS500 Alternator Regulator. However, in more sensitive installations where the management of the alternator Field Drive at low RPMs is critical, additional system reliability can be achieved by defining the IDLE RPMs value to be used in all calculations. In extreme installations (very small engine with large efficient alternator), Idle RPMs may be defined artificially high; doing so will cause the regulator to increase its pull back of Field Drive during low RPM operations.

Set = 0 (DEFAULT) to enable 'auto' determination of Idle RPM.

Warmup Delay: < WHOLE NUMBER (15 → 600) > Hold-off period when regulator is 1st powered on before it will begin to apply a load to the engine. This is the number in seconds of delay the regulator will remain in PRE-RAMP mode before moving into RAMP mode. Default = 30 seconds.

Required Sensors: < WHOLE NUMBER (0 → 255) > Many capabilities depend on the presence of sensors. Battery compensation requires the presence of a battery temperature sensor; Alternator Temperature regulation requires the presence of an alternator temperature sensor. If one or more of these sensors are not installed, or fail during operation, results could be less than desired. As a precaution against this, *Required Sensors* allows the identification of critical sensors, and if any of them are missing or fail the regulator will take action to reduce demands placed on the system.

Required Sensors allows the identification of critical sensors. It is a number created by summing up the value associated with each potential critical sensor. For example: if you wished to indicate the Alternator and Battery temperature sensors are critical, you would enter 3 (1+2). The value of 0 disables critical Required Sensor checks and the regulator will utilize other existing fall-back modes.

Sensor	Value	Default Action of missing sensor
Alternator Temperature Sensor	1	Enable Half-Power mode
Battery Temperature Sensor	2	Force to FLOAT mode
Battery Current **	4	Force to FLOAT mode (See note**)
CAN RBM	8	Disable charging, Feature-in may be used to Force to Float when CPE #8 is selected
Force FAULT override	128	Overrides 'Default' action and forces regulator into FAULT mode.

Table 1 - Required Sensor Encoding

If at any time one of the Required Sensors are identified as failed or missing the LED will flash its normal patterns, but in RED. In addition the feature-out port / dash-lamp it will also flash out error codes during a fault.

The WS500 Alternator Regulator may also be configured to cause a non-recoverable FAULT condition, overriding the default actions listed in Table 1 by adding 128 to the summed number. In the prior example of Bat and Alt sensors being critical, sending 131 instead of 3 will cause the regulator to FAULT if either is noted as missing or fails.

Battery current / Battery Temperature values arriving via the CAN from a remote battery master or BMS may be used to satisfy the Bat Temp and or Bat Current checks in *Required Sensors*. In addition, by selecting *CAN RBM* a check for the entire presence of a remote battery master or BMS may be required as well.

Note** When not receiving Battery Current via the CAN, and utilizing the shunt for measurements, it is difficult to determine if an Amp Shunt has failed vs. a true reading 0A of current. Because of this, the WS500 Alternator Regulator will delay check for the presence of a working Current Shunt until after Bulk has been completed. A current reading of greeter then 5A at any time during BULK phase will be used as an indicator that the current shunt is present and working, as will the arrival of battery current measurements via the CAN. Once this determination is made no additional checks will be made – as a valid operation condition for the regulator may at times have a battery current of 0A (example, when actively regulating current to 0A in FLOAT mode).

\$SCA: will reply with “AOK;” if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

Example - Large alternator powered by large main engine:

- If not already done, set regulator’s name
- Disable Idle adaptive pullback (alternator has massive cooling capability, and engine has sufficient reserve)
- Adjust amp shunt to 250A/75mV
- Define engine idle @ 550 RPMs
- Must have Alternator Temperature sensor present, else cause regulator to FAULT
- Leave other values as default (See *Appendix C*:)

```
$SCN:0,MainsAlt,5555@  
$SCA:0,95,1.0,0.75,0.50,0,0,0,3333,0,550,60,129@  
$RBT:@
```

Example - Detailed configuration. Large alternator powered by large main engine, 1500AH industrial FLA battery:

- Set name to MainsAlt
- Acceptance @ 14.4v until acceptance current is less than 1% of capacity, 8hr max.
 - *8hr Time limit is set as fall-back in case of battery current sensing failure.*
- 13.2v float - revert back if 2% of capacity is removed
- 15.3v Equalize, 3Hr duration.
- No overcharge nor post-float phases.
- Allow Alternator to operate up to 105c
- 30mV temp comp

```
$SCN:0,MainsAlt,4449@
$SCA:0,105,1.0,0.75,0.50,0,0,0,3333,0,550,30,0@
$CPA:7 14.4,480,5,0@
$CPO:7 0,0,0,0@
$CPF:7 13.2,-1,0,0,-10,12.7,0@
$CPP:7 0,0,0,0.0@
$CPE:7 15.3,0,180,0@
$CPB:7 0.030,0,-45,45,0,-99,-99,0,0@
$SCO:7,3,1,0@
$RBT:@
```

Example: Shorted warm-up delay to 15 seconds, leaving the rest of the configuration at the default values:

```
$SCA:0,90,1.0,0.75,0.50,-1,0,0, 10000,0,0,15,0@
```

§SCT: - Changes TACHOMETER parameters in System Configuration table



Update calibration ratios and parameters associated with alternator driven tachometers. It should be noted the regulator will function correctly without changing any of these parameters; you need only change them if you wish to estimate the RPMs of your engine to be reported by the WS500 Alternator Regulator.

§SCT: <Alt Poles>, <Eng/Alt drive ratio >, <Tach Min Field>, <ForceTM>

Alt Poles: <WHOLE NUMBER (2 → 25)> Number of poles in the alternator.

Eng/Alt Drive Ratio: <FLOATING POINT NUMBER (0.5 → 50)> Enter the ratio your engine drive pulley diameter vs. the alternator drive diameter. Example, if your engine has a 7" drive pulley, and the Alternator has a 2.6" drive pulley, then enter: 2.6923 (7.0 / 2.3)

Tach Min Field: <WHOLE NUMBER (-1 → 30)> This is the % value the PWM will be kept at as the minimum drive when the DIP switch has selected TACH MODE. *BE VERY CAREFUL* with this value as it will set the floor in which the alternator is driven. If that floor is too high, it will prevent the regulator from 'regulating', burning out the battery. This is the actual PWM value sent to the field drive; though it is capped at 30% the full hardware PWM.

Set *Tach Min Field* = -1 to enable auto-determination. The WS500 Alternator Regulator will monitor the stator signal and when it becomes stable will use that PWM drive value as the floor. Alternatively, set this = 0 to disable any tach field drive even if the DIP switch is turned on.

If a *Tach Min Field* value is set (any value greater than 0), and TACH-MODE is enabled, the regulator will use this value as a minimal field drive %. Even during the warm-up period. You may have to experiment with this value to get one which matches your system, taking care not to make it too great, as that could cause issues with overcharging of your battery.

If a fixed *Tach Min Field* value is set, the regulator will also hold in the warmup / idle phase until it is able to stably see RPMs, indicating the engine is running. This will prevent the field from being driven any harder in the case of the regulator is powered on via ENABLED, but the actual engine is not running. Do take note though: if the stator wire is not connected (or has failed), and/or the *Tach Min Field* % drive value is too low, the regulator will remain in warm-up mode, 'appearing' to have failed when in fact it was been instructed to wait until it can see a stator signal...

Note: Effective with Firmware version 1.0.0 and above *Tach Min Field* has been changed from a 'RAW PWM' value to a % of full field ratio. Take note of this change and make adjustment during future use.

ForceTM: <0, or 1> Setting the value to 1 will force on Tach-mode, independent of the Tach-mode DIP switch (if equipped). 0 will or cause the regulator follow the DIP-Switch, or disable Tach-Mode if no dip-Switch is present. (0 = DEFAULT)

\$SCT: will reply with "AOK;" if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

\$SCT: will not be recognized if system has been locked-out via the \$SCO: command.

\$SCO: - Override features

Overrides the DIP Switches for Charge Profile and Battery Capacity selection. This command also allows the selection of auto detect for system voltage (12v, 24v, 48v), or forcing a fixed defined target system voltage.

\$SCO: <CP_Index>, <BC_Index>, <SV_Override>, <Lockout>

CP Index: <WHOLE NUMBER (0 → 8)> Which Charge profile entry should be used? (1..8). Set = 0 to use DIP switches for selection.

BC Index: < FLOATING POINT NUMBER (0.0 → ±10.0)> Which Battery Capacity Multiplier entry should be used against normalized 500Ah battery? (1..4). Set = 0.00 to restore selection to DIP Switch value. **It is possible to also receive the battery capacity via CAN. In normal cases the BC Index will be overridden by a CAN received value. However, if you do not wish the BC Index to be overridden by CAN received values; configure BC Index with a negative value. The multiplier will have the same effect, the sign of the value only indicates if the BC Index should also override any CAN received value or not. Positive = yes, allow a CAN supplied value to override; negative = no, ignore both the DIP switch and any value is supplied via the CAN.**

SV Override: < FLOATING POINT NUMBER (0.0 → 4.0)> Enable (by setting = 0.0) or override the auto system voltage detection feature by defining the SV multiplier to be used. Though Auto detect is a nice feature, being able to fix the system voltage can improve reliability and allow support for battery voltages which are not a whole number multiple of the '12v' normalized battery used in the CPE tables. The following table shows some common values which may be used:

SV_Override value	Forced System Voltage	Charge Profile VOLTAGE Adjustment Factor
0	Auto	Auto
1	12v	1x
2	24v	2x
2.67	32v	2.67x
3	36v	3x
3.5	42v	3.5x
4	48v	4x

(Set SV Over-ride = 0 to restore auto-selection of 12v, 24v, or 48v system voltages)

Lockout: <WHOLE NUMBER (0 → 2)> Security feature: Restricts ability to perform changes and/or provide input to the regulator which can impact how the Alternator charges the battery. **BE CAREFUL:** Once lockout is enabled (value other than 0), it can **ONLY** be cleared by doing a hardware based master reset, or re-flashing the firmware with changes to the EEPROM keys. No other command, not even \$MSR: will be able to clear a non-zero lockout. ***Be especially careful with lockout=2, as the ONLY way to recover the regulator from this state will require special programming hardware probes.***

0 = No locking out.

1 = Prevent any configuration changes.

2 = Prevent any configuration changes – may NOT be cleared via Feature-in method.

\$SCO: will reply with “AOK;” if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

\$SCO: will no longer be recognized once it has been locked-out.

Example - Configure to override DIP switches and positively define system voltage and battery capacity:

- If not already done, set regulators name
- Use CPE#3 (FLA#2 – large batteries)
- 1500AH battery (BC Index = 3)
- 12v system (SV Override = 1)
- Lockout NOT enabled (Allows continued changes)

```
$SCN:0,MainsAlt,5555@
```

```
$SCO:3,3,1,0@
```

```
$RBT:@
```

§SCN: - Changes NAME (and PASSWORD)



Update Name and Password configuration. The name is used by the CAN subsystem to identify *this* regulator. To protect the user and reduce the possibility of hacking on devices equipped with Bluetooth, the ability to make any configuration changed is disabled until the name and password has been updated. The 1st time the §SCN: command is used the alternator must not be actively charging, AND all the DIP switches must be in the ON state. Once the Regulators Name & Password has been initial changed they may be updated at a later time independent of the alternator state and/or DIP switch state. Resetting the regulator to Factory default configuration, or clearing the System Configuration will require you to again change the regulators name/password before any other commands will be recognized.

This command will also clear any prior Bluetooth associations saved in the Bluetooth module – you will need to re-connect.

§SCN: <RESERVED>, < Reg Name >, <Reg Password>

RESERVED <WHOLE NUMBER (0, or 1)> Set = 0;

Reg Name: <STRING (up to 18 characters, no spaces, comma, or '@') > Name used for CAN ID. If you have twin engines, you might wish to set these to descriptive names.

Reg Password: <STRING (up to 18 characters, no spaces, comma, or '@') > .

§SCN: will reply with "AOK;" if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the §RBT: command after all changes have been made.

§SCN: will not be recognized if system has been locked-out via the §SCO: command.

\$SCR: - RESTORES System Configuration table to default



Restores System Configuration values to original factory default values.

\$SCR: will reply with "AOK;" if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

\$SCR: will not be recognized if system has been locked-out via the \$SCO: command.

\$CCN: - Change parameters in the CAN Configuration table



Configure the CAN Configuration Table for this regulator

\$CCN: <Battery Instance Override>, <Device Instance >, <Device Priority>, <AllowRMB?>, <ShuntAtBat?>, <Enable-OSE?>, <Enable-NMEA2000?>, <Enable_ALT_CAN?>, <Engine ID>,<BitRate>,< DC_ Disconnected _VBat>, <Aggregate_BMS>

Battery Instance Override: <WHOLE NUMBER (0 → 100)> What battery instance is this device associated with? (1..100). Set = 0 to use DIP switches for selection.

Device Instance: <WHOLE NUMBER (1 → 13)> Which instance of charging devices is this? Allows unique identification of charging sources. *Device Instance is mostly used by external displays. It is considered good practice to give each regulator a unique Device Instance number, however if the installer fails to do that the WS500 will automatically adjust the Device Instance number in case of a conflict.* (Default = 1)

Device Priority: <WHOLE NUMBER (1 → 250)> What is the relative priority of this charging device?

A key value of the OSEnergy protocol is the ability to prioritize charging sources. This value is what is used to decide a given charging sources priority. If the needs of the associated battery (and any additional loads) can be met by higher priority charging sources the regulator will reduce its output to 0A. However, if the battery/load needs cannot be meet the regulator will deliver current to its limits as needed. If there are two or more charging sources with the same priority, battery /load needs will be split between them. (Useful in dual engine installations to balance loads between both engines). (Alternators Default=70)

Device priority is also used to decide who should act as the Remote Battery Master, or the overall coordinator in the system to assure all charging devices are working towards the same goal. If *AllowRBM?* Is enabled, the WS500 Alternator Regulator will assume the RBM role if no other higher device exists. This can be useful in simple installations where no Battery Monitor is installed or as a fall-back for a failed battery monitor.

Allow RBM?: <WHOLE NUMBER (0, or 1)> Should the WS500 Alternator Regulator attempt to act as the Remote Battery master?

0 = Do not allow the regulator to assume the RBM role.

1 = Allow the regulator to potentially assume the RBM role. (Default)

Shunt At Bat?: <WHOLE NUMBER (0, or 1)> Is the shunt connected to the Battery? Used during RBM mode to know if we are seeing alternator or battery current. 0=no, 1=yes(default)

Enable OSE?: <WHOLE NUMBER (0, or 1)> Should the WS500 Alternator Regulator send and receive OSEnergy (RC-V) status and coordination messaged via the CAN bus? 0=no, 1=yes(default)

There may be some simple installations where one wishes to use the WS500 Alternator Regulator to only broadcast status to NMEA2000 devices, and the OSEnergy messaging (RV-C standard) causes issues with some existing NMEA2000 devices. Do note that disabling OSEnergy mode will remove many of the systems benefits such as coordinated / prioritized charging, simplified remote instrumentation, and more.

Enable NMEA2000?: <WHOLE NUMBER (0, or 1)> Should the WS500 Alternator Regulator send out NMEA-2000 like status messages via the CAN bus? 0=no, 1=yes(default)

Enable_ALT_CAN: <WHOLE NUMBER (0 → 255) > Though RBMs supporting RV-C/OSEnergy protocols provide the best level of integration, there are other CAN based BMS and remote sensors which may be utilized. *Enable_ALT_CAN* is a way to signal which of those additional protocols to support. Note that some protocols are able to provide full RBM direction, while others only offer a limited amount of remote sensing. It is up to the installer to assure which messages are supported by the attached device and if the resulting system integrations meet expectations and needs. A value of 0 (default) will disable any additional CAN protocol support, and it should also be noted that a given device may require other configuration changes such as adjusting CAN baud rate, and even disabling some optional protocols supported (e.g., NMEA2000)

Protocol	Value	Features
NMEA2000 RAT	1	Battery current and temperature via NMEA2000 battery monitor.
SMA	2	BMS devices supporting 11-bit ‘Sunny’ SMA type messages. Aka, ‘Victron BMS’ protocol
SMA w/“Zero Output Technology” enabled.	3	Enabled active regulation of battery current to OA (and passed voltage goal) instead of disabling alternator with so signaled from BMS.
VE.can	4	Support for Victron VE.can via VREGs messages
VE.can w/“Zero Output Technology” enabled.	5	VREGs messages with active regulation of battery current to OA.

Table 2 – Alternative CAN protocols support

Note: Prior to Firmware v 2.1.0, *Enable_ALT_CAN* was a single value of 0 or 1 and used to specify only NMEA200_RAT support. Refer to Appendix A: CAN enabled BMS for additional details.

Engine ID: <WHOLE NUMBER (0 → 250) > Used to associate the regulator with the engine it is mounted on. Specifically with regards to RPMs. The regulator monitors for a matching J1939 engine RPM (PGN: 61444) and will use it instead of measured stator RPMs. Also, if NMEA2000 messages are enabled and the regulator is able to measure RPMs – NMEA2000 PGN: Engine parameters rapid (#127488) will be sent with the RPMs indicated as being associated with this Engine ID. Default ID = 0.

BitRate: <WHOLE NUMBER (0 → 4)> If supported by hardware, allows CAN communications rate to be changed from the default 250Kbps.

Value	Rate
0	Default (250Kbps)
1	100Kbps
2	125Kbps
3	250Kbps
4	500Kbps

DC_Disconnected_VBat: <FLOATING POINT NUMBER (0.0 → 20.0)> If a DC-Disconnect command is received via the CAN bus, the charging source will normally be placed into a Disabled mode. However, it may also be optional placed into a CV state using this voltage. A value of 0.0 (default) causes transition to DISABLED mode.

Aggregate BMS: <WHOLE NUMBER (0 → 10)> Some installs utilize more than one BMS in a given battery bank. This may be done to support slightly different locations (ala, port and starboard side of a vessel due to space constraints and/or weight distribution), or more commonly to increase maximum available current by paralleling BMSes. In doing so, not all BMS devices will perform aggregation, instead they will present themselves as different Battery Instance ID's, despite being connected to the same physical DC bus. *Aggregate BMS* allow support of this type of install by monitoring for any BMS or potential RBM with not only the configured *Battery Instance* number (See above), but also up to 10 Battery Instances. From the base *Battery Instance* up to *Battery Instance + Aggregate BMS*. Example, if *Battery Instance* is set for 2 (either via the DIP switches, or *Battery Instance Override* above) and *Aggregate BMS* is set for 5, then ANY battery Instance from 2 to 7 will be consider all part of the same 'battery' and aggregated by the WS500.

Aggregation will sum up the battery current, and capacity; will monitor for any given battery device coming on line or going offline. The request charge state will be the least aggressive (e.g., if ONE online battery asked for Float, then the WS500 will enter Float mode even if the remaining batteries are asking for Bulk). The battery with the most extreme temperature (Hot or cold) will be used to make temperature related decisions, and if a given battery goes off line the WS500 will continue to monitor and respond to the remaining batteries. Once if ALL batteries go offline the WS500 will enter a fault state.

Aggregate BMS is an advanced capability, and it is up to the installer to test and assure the overall system responds in an expected way. Note that if enabled, Remote Voltage Sensing is disabled and the WS500 MUST be installed with the voltage sensing wires attached to an appropriate point in the battery bank. At present, the following BMS's have been proofed with aggregated:

- MG Energy BMS
- Lithionics NeverDie™ BMS

A value of 0 (default) will disable any battery aggregation.

\$CCN: will reply with "AOK;" if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

\$CCR: - RESTORES CAN Configuration table to default

Restores CAN Configuration values to original as-compiled (default).

\$CCR: will reply with "AOK;" if the command was processed successfully. However, to assure the changes are STORED and used the regulator must be reset using the \$RBT: command after all changes have been made.

\$CCR: will not be recognized if system has been locked-out via the \$SCO: command.

\$MSR: - RESTORE all parameters (to as defined at program compile time).

Restores all configurable parameters to the as-compiled (default) values. This is a combination of the \$SCR: , \$CCR commands, the \$CPR:n commands for all Charge Profile Entry tables, and \$RBT; command. Plus, any calibrations done are cleared. Alternator will RESET after this command is completed.

\$MSR:

Note \$MSR is disabled if the Regulator has been locked out via the \$SCO command. In this case, you will need to do to do a full system reset via the Feature In connector. If successful, the regulator will reply with "AOK;" and then reboot.

\$MSR: will not be recognized if system has been locked-out via the \$SCO: command.

\$EDB: - Enable DeBug serial strings



Will cause regulator to start sending \$DBG; strings via ASCII communication ports. If it is powered down, or reset (ala a Fault, or by receiving a command string that causes a reset), the regulator will restore to its default handling of \$DBG: strings.

\$EDB:

\$RBT: - ReBooT system



Will cause regulator to reset. This is useful to load any changes from saved Flash memory into the regulator for its use.

\$RBT:

\$RBT: will not be recognized if system has been locked-out via the \$SCO: command.

It is strongly suggested that you use the \$RBT command after you have finished making changes to the Alternators configuration via other ASCII commands. This will restart the regulator, allowing those changes to be recognized – but more important some hardware needs the \$RBT: command as a signal to actually save requested changes in non-volatile memory. If your device using what is known as EEPROM-Emulation, any changes you make will not be saved until you issue the \$RBT: command.

\$FRM: - Force Regulator Mode



This command (with its parameters) will force the regulator to change its current mode to the one indicated. Once forced into a mode the regulator will continue to manage the system accordingly, even if this means the regulator immediately exits the forced mode. For example, if you force the regulator into Float mode, but the Amps being taken from the battery exceed the `exit_float` criteria, the regulator will return to the Bulk phase.

`$FRM:<Mode>`

Mode: <Character> The ASCII character *immediately* following the ':' will be used to force the alternator mode. Character must match EXACTLY the following (including case), must be IMMEDIATELY after the ':', but may be followed by any number of additional characters.

- B = Force into BULK mode.
- A = Force into ACCEPTANCE mode.
- O = Force into OVER-CHARGE mode.
- F = Force into FLOAT mode.
- P = Force into POST-FLOAT mode.
- E = Force into EQUALIZE mode.
- C = Force into CONFIGURATION mode

Any other character will be ignored and no change will be made. If the active Charge Profile has 'disabled' a given phase, the regulator will immediately exit that phase – even if it is 'forced' into it, and switch to the next appropriate phase. Also note that if the exit criteria of a forced-mode phase is met, the regulator will again exit that phase quickly. In such cases the mode may be changed before the next \$AST string is sent.

Configuration Mode is a special mode where the regulator will listen and process all CAN and ASCII communications, but it will not actively drive the alternator nor doing error checking. It can be useful if you wish to do extensive configuration of the regulator without worrying about overheating the field, or have faults reported due to not yet in place sensors. As well as allowing simpler bench-top configuration w/battery under-voltage faults occurring. Once entered, the only way to exit CONFIGURATION mode is to \$RBT: or power-cycle the regulator.

Examples:

- \$FRM:B → Forces regulator into BULK mode
- \$FRM:Bulk → Forces regulator into BULK mode
- \$FRM:Bob@ → Forces regulator into BULK mode (Note use of '@' as needed with Arduino IDE terminal)
- \$FRM:b → Ignored (lower case 'b')

No pre-existing condition check is made when receiving these mode change commands. For example, normally you would be able to enter Equalize mode only if the regulator was already in Float or Post-float mode. However, the \$FRM: command can force the regulator into Equalize mode directly from any state, including Bulk or Ramping. (Do remember: as noted above - if conditions are such, it may not stay in Equalize very long.)

APPENDIX A: CAN ENABLED BMS

The ability for the WS500 Alternator Regulator to communicate with BMS devices using the CAN (Control Area Network) provides for a higher level of integration and reliability than is possible using only legacy methods (e.g., Charge Enable wire), as well as installation simplification via remote instrumentation of battery current, voltage, and/or temperature (BMS support dependent). This appendix provides details of optional CAN protocols supported by the WS500 regulator. Refer to the command \$CCN: on page 45 and how the 'Enable_ALT_CAN' field is defined.

WARNING: Not all BMS devices provide advanced notification of pending DC disconnect. It is up to each installer/designer to verify the details of the BMS and its integration with the Wakespeed WS500 Advanced Alternator Regulator to assure a reliable system. It is suggested that at minimum a 2 second warning be issued by the BMS before any actual physical disconnect occurs.

CAN based BMS integration can be a challenging concept at first, though the end result does allow for a simpler and more robust overall system install. If the installer has any questions about what is contained in here, or the suitability of a given BMS, send an Email to: support@TJCMicro.com

NMEA2000_RAT: Should the WS500 Alternator Regulator look for a NMEA2000 device to supply remote sensing of battery Amperage and Temperature via PGN: 127508? To reduce confusion in the NMEA2000 network, if *Enable_NMEA2000_RAT* is set = yes, and the regulator notices any other device supplying this information it will stop sending out its own versions of PGN: 127508.

Example product(s) include:

- Maretron DCM100

SMA: The ‘Sunny’ SMA CAN protocol is an 11-bit (2.0A) message commonly used in the Solar and Electric Vehicle industry. It allows for a BMS to tell charging sources to start, stop, and which goals to utilize. Often known as the SMA protocol, it is also reflective of the Victron 11-bit BMS protocol, though at a different CAN speed. Some representative BMS’s which use this protocol includes:

- MG Energy (<https://www.mgennergysystems.eu/>) **
- REC BMS (<https://www.mgennergysystems.eu/>) **
 - MUST use ‘Victron’ variant of this BMS
- Orion Jr (<https://www.orionbms.com/>)
 - Select ‘Victron Inverter’
- SIMP BMS (<https://github.com/tomdebree/SimpBMS>)

** Note: These BMS’s have been tested in the lab and/or in field installations. Others may have only been confirmed via specifications. Contact support@wakespeed.com if you have any questions on a specific BMS.

The 11-bit CAN messages will be ‘converted’ into an OSEnergy type device with a fixed CAN-ID of 69 and a priority of 120. Aggregation is not supported. The following CAN messages are utilized. (Refer to your BMS guide for details of these messages)

- 0x351: Battery Charge Voltage, Charge Current Limitation
- 0x356: Battery Current, Battery Temperature
- 0x35A: Alarm/Warning

Both 0x351 and 0x356 must be broadcasted at a minimum rate of 1,000mS to allow the WS500 to recognize a device and begin following its directions. Once doing so, if *Battery Charge Voltage* and *Charge Current Limitation* (Message 0x351) are both non-zero the WS500 will enter CV/CC mode using those goals. A zero value for one or both of those will cause the WS500 to enter standby. Optionally, if *SMA w/Zero Output Technology* is selected a zero goal value for current with a non-zero value for voltage will cause the WS500 to actively regulate battery current to 0A, while leaving the alternator able to supply energy to ongoing house loads.

Message 0x356 may be used to remotely supply *Battery Current* readings to the WS500, reducing the need to install a physical current shunt. In addition, *Battery Temperature* measurements will be used in combination with the presently active CPE to enforce hard charging limits based on high and/or low temperatures, even if the BMS continues to request charging.

0x35A: The Alarm/Warning message will be monitored for any active indication. Upon receiving a Warning indicators, the WS500 will stop charging and perform a recoverable fault (regulator will restart). However if a Fault flag is received, the WS500 will enter a non-recoverable hard fault.

Different BMS devices behave differently, but the best practice is if the BMS sends out a warning and/or fault 2 seconds before doing a physical disconnect to prevent voltage spikes in the system from an uncoordinated disconnect.

VE.Can: VE.Can is an extended addressing (2.0B) CAN message supported by Victron™ and allows for a more capable level of integration between a BMS and the WS500. It is also known as `VREGS`. These messages are sent using the J1939 'proprietary' message protocol via message PGN 0xEF00. Some BMS devices (example the MG Energy series BMS) support both 11-bit and 29-bit messaging, in such cases it is preferred to utilize the 29-bit messages. Some representative BMS's which use this protocol includes:

- MG Energy (<https://www.mgenergysystems.eu/>) **

** Note: These BMS's have been tested in the lab and/or in field installations. Others may have only been confirmed via specifications. Contact support@wakespeed.com if you have any questions on a specific BMS.

The VREG CAN messages will be 'converted' into an OSEnergy type device using the original CAN node address and with a fixed priority of 120. Aggregation is supported and may be enabled via the \$CCN: command. Within the VREG specification the following registers are utilized: (Refer to your BMS guide and/or Victron VREGS documentation for additional details)

- 0x371: BMS State
- 0x390: Battery Charge Voltage
- 0x391: Charge Current Limitation
- 0xED8F: Battery Current
- 0x0FFF: SOC
- 0x0386: Battery Temperature
- 0x1000: Battery Capacity

Registers 0x371, 0x390, 0x391, and 0xED8F must be received with at least one every 500mS. To help facilitate this, the WS500 will issue a request for Battery Current every 500mS. Not only to allow the heart-beat to continue, but also to enable remote instrumentation of Battery Current.

Once a BMS enters the READY state, if it transitions out of that state the WS500 will consider that as an Alarm condition and enter a hard fault.

As with the SMA protocol, when *Battery Charge Voltage* and *Charge Current Limitation* are both non-zero the WS500 will enter CV/CC mode with those goals. A zero value for one or both of those will cause the WS500 to enter standby. Optionally, if “w/Zero Output Technology” is selected a zero goal value for current with a non-zero value for voltage will cause the WS500 to actively regulate battery current to 0A, while leaving the alternator active to supply energy to ongoing house loads.

APPENDIX B: CAN MESSAGES

The WS500 Alternator Regulator contains a Control Area Network (CAN) subsystem. The purpose of this network is to allow communications of status, configuration, and coordination of charging in a systems view.

The WS500 Alternator Regulator utilizes a mixture of open source standards, and reverse engineered standards including:

- OSEnergy – (<https://github.com/OSEnergy/OSEnergy>) Open Systems Energy initiative: Overriding specification defining communication hardware and protocols allowing for coordination of charging devices.
- J1939 – SAE standard providing basic coordination of nodes and communications of messages
- NMEA-2000 – Marina orientated status messages. Closed specification built upon J1939 which has been reverse engineered.
- RV-C -- (RV-C.com) True open source specification targeting primary Recreation Vehicle industry, but extended to include many needed communications to support the OSEnergy initiative.
- Others: A few selected 2.0A (11-bit addressing) messages are supported as documented in \$CCN on page 45.

CAN messages summary

The following summarizes the pgn's sent and/or received via the WS500 Alternator Regulator over the CAN bus. Items in **BOLD** are utilized by the WS500 (Sent or received) – though the actual messages may be impacted if a subsystem is enabled as well as if the WS500 is operating as the RBM for the system.

'NMEA-2000 type' messages

```

/*****
// NMEA2000-DC Detailed Status - PGN127506
// Input:
// - SID Sequence ID. If your device is e.g. boat speed and heading at same time, you can set
// same SID for different messages to indicate that they are measured at same time.
// - DCInstance DC instance.
// - DCType Defines type of DC source. See definition of tN2kDCType
// - StateOfCharge % of charge
// - StateOfHealth % of health
// - TimeRemaining Time remaining in minutes
// - RippleVoltage DC output voltage ripple in V

```

```

/*****
// NMEA2000-Battery Configuration Status -- PGN127513
// Note this has not yet confirmed to be right. Specifically Peukert Exponent can have in
// this configuration values from 1 to 1.504. And I expect on code that I have to send
// value PeukertExponent-1 to the bus.
// Input:
// - BatteryInstance BatteryInstance.
// - BatType Type of battery. See definition of tN2kBatType
// - SupportsEqual Supports equalization. See definition of tN2kBatEqSupport
// - BatNominalVoltage Battery nominal voltage. See definition of tN2kBatNomVolt
// - BatChemistry Battery See definition of tN2kBatChem
// - BatCapacity Battery capacity in Coulombs. Use AhToCoulombs, if you have your value in Ah.
// - BatTemperatureCoeff Battery temperature coefficient in %
// - PeukertExponent Peukert Exponent
// - ChargeEfficiencyFactor Charge efficiency factor

```

```

/*****
// NMEA2000-Battery Status - PGN127508
// This PGN will not be sent if ENABLE_NMEA2000_RAT is set = YES.
// Input:
// - BatteryInstance BatteryInstance.

```

```
// - BatteryVoltage      Battery voltage in V
// - BatteryCurrent     Current in A
// - BatteryTemperature Battery temperature in Å,Å°K. Use function CToKelvin, if you want to use Å,Å°C.
// - SID                 Sequence ID.
```

```
/******
```

```
// Engine parameters rapid - 127488L
// Input:
// - EngineInstance      Engine instance.
// - EngineSpeed        RPM (Revolutions Per Minute)
// - EngineBoostPressure  in Pascal
// - EngineTiltTrim
```

```
/******
```

```
// NMEA2000-Charger Status - PGN127507
// Input:
// - Instance            ChargerInstance.
// - BatteryInstance    BatteryInstance.
// - Operating State    see. tN2kChargeState
// - Charger Mode       see. tN2kChargerMode
// - Charger Enable/Disable boolean
// - Equalization Pending  boolean
// - Equalization Time Remaining double seconds
```

RV-C messages (in support of OSEnergy standard)

```

/*****
// DC Source Status 1
// Input:
// - Instance           DC Instance (bus) ID.
// - Device Priority   Relative ranking of DC Source
// - DC Voltage       0..3212.5v, in 50mV steps
// - DC Current       -2M..+2MA, in 1mA steps (0x77359400 = 0A)

/*****
// DC Source Status 2
// Input:
// - Instance           DC Instance (bus) ID.
// - Device Priority   Relative ranking of DC Source
// - Source Temperature -273 to 1735 Deg-C in 0.03125c steps
// - State of Charge     Batteries: % SOC; DC Charging sources: Current % output.
// - Time Remaining      Estimated number of minutes until SOC reaches 0%

/*****
// DC Source Status 4
// Input:
// - Instance           DC Instance (bus) ID.
// - Device Priority   Relative ranking of DC Source
// - Desired Charge Mode Charging mode / state being requested.
// - Desired DC Voltage Target voltage for chargers to deliver 0..3212.5v, in 50mV steps
// - Desired DC Current Target current for all chargers to deliver combined -1600A..1612.5A, in 50mA steps (0x7D00 = 0A)
// - Battery Type

/*****
// DC Source Status 5
// Input:
// - Instance           DC Instance (bus) ID.
// - Device Priority   Relative ranking of DC Source
// - DC Voltage       High precision value in 1mV. Useful for remote instrumentation
// - VDC ROC             Rate-of-change (dV/dT) in mV/s -- 32000 = 0 mV/s

/*****
// DC Source Status 6 - 1FEC7h
// Input:
```

```

// - Instance           DC Instance (bus) ID.
// - Device Priority   Relative ranking of DC Source
// - HV Limit Status   Reached upper operational voltage range?
// - HV Limit Disconnect Safety disconnect?
// - LV Limit Status     Reached lower operational voltage range?
// - LV Limit Disconnect Safety disconnect?

//*****
// DC Source Status 11 - 1FEA5h
// Input:
// - Instance           DC Instance (bus) ID.
// - Device Priority   Relative ranking of DC Source
// - PwrOnOff          Status of main Battery Switch / Contactor (TRUE = connected)
// - ChrgOnOff        Status of Charge Bus switch (TRUE = connected)
// - ChrgDet             Has a charge source been detected?
// - ResvStat           Is battery running on its 'reserved capacity'?
// - BatAHCap         Battery capacity in Ah's
// - DCPower            Watts being received or delivered to/from battery (as opposed to AMPS in DC_STATUS_1)

//*****
// Charger Status - 1FFC7h
// Input:
// - Instance
// - Charge Voltage     0..3212.5v, in 50mV steps
// - Charge Current    -1600..+1512.5 in 50mA steps (0x7D00 = 0A)
// - % max current
// - Operating State   (Bulk, float, etc)
// - Default PO state
// - Auto Recharge
// - Force Charged
*/

//*****
// Charger Status2 - 1FEA3
// Input:
// - Instance           Instance of charger
// - DC Source Instance DC Instance (bus) ID associated with
// - Device Priority   Relative ranking of DC charging Source
// - DC Voltage       0..3212.5v, in 50mV steps
// - DC Current       1600..+1512.5 in 50mA steps (0x7D00 = 0A)
// - Temperature      -40..210 in deg-C, in 1C steps

```

```

/*****
// Charger Configuration Status - 1FFC6h
// Input:
// - Instance
// - Charging Algorithm
// - Controller Mode
// - Battery Sensor Present
// - Charger AC Line          Line 1 or 2 (AC Chargers only)
// - Linkage Mode
// - Battery Type
// - Battery Bank Size        0..65,530 Ah, 1Ah increments
// - Maximum charging current 0..250, 1A increments
*/

```

```

/*****
// Charger Configuration Status2 - 1FF96h
// Input:
// - Instance
// - Max Charge Current %
// - Max AC current %         Of attached line          (AC Chargers only)
// - Shore Breaker Size       0..250, 1A increments (AC Chargers only)
// - Default Batt Temp
// - Recharge Voltage         0..3212.5v, in 50mV steps
*/

```

```

/*****
// Charger Configuration Status3 - 1FECCh
// Input:
// - Instance
// - Bulk Voltage             0..3212.5v, in 50mV steps
// - Absorption Voltage       0..3212.5v, in 50mV steps
// - Float Voltage           0..3212.5v, in 50mV steps
// - Temp Comp               mV/K
*/

```

```

/*****
// Charger Configuration Status4 - 1FEBFh
// Input:
// - Instance
// - Bulk Time          0..65,530min in 1min steps
// - Absorption Time    0..65,530min in 1min steps
// - Float Time         0..65,530min in 1min steps
*/

/*****
// Charger Equalization Status - 1FF99h
// Input:
// - Instance
// - Time Remaining     0..65,530min in 1min steps
// - Pre-Charging
*/

/*****
// Charger Equalization Configuration Status - 1FF98h
// Input:
// - Instance
// - Equalization Voltage 0..3212.5v, in 50mV steps
// - Equalization Time    0..65,530min in 1min steps
*/

/*****
// Terminal - 17E00h
// Input:
// - Source / Destination
// - Count              0..8
// - Characters         Buffer with up to 8 characters
*/

```

```
/*  
// ISO Diagnostics message - 1FECAh  
// Input:  
// - On / Off  
// - Active / Standby  
// - DSA Default Source Address (Standard fault codes)  
// - SPN Service Point Number (Device Specific)  
// - FMI Failure Mode Identifier  
// - Occurrence Count  
// - DSA Extension  
// - Bank Select  
*/
```

APPENDIX C: DETAILS OF CPE (CHARGE PROFILE ENTRIES)

The following will give more details on how each parameter impacts battery charging.

```
Typedef struct {                                     // Charging Profile Structure -- how will we CHARGE a battery?

    float      ACPT_BAT_V_SETPOINT;                 // Set point for Ramp, Bulk and Acceptance battery voltage.
                                                    // Alternator will transition from BULK mode into Accept Mode when this voltage is
                                                    // reached, and then start the Accept Duration counter.

    uint32_t   EXIT_ACPT_DURATION;                 // Stay in Accept mode no longer then duration in mS (Set = 0 to disable Acceptance phase
                                                    // and move directly to OC or Float mode)

    int        EXIT_ACPT_AMPS;                    // If Amps being delivered falls to this level or below, exit Accept mode and go to next
                                                    // Set ExitAcptAmps = 0 to disable Amps based transition and only rely on
                                                    // EXIT_ACPT_DURATION timeout.
                                                    // Set ExitAcptAmps = -1 to disable Amps based transition and rely on
                                                    // EXIT_ACPT_DURATION timeout
                                                    // or ADPT_ACPT_TIME_FACTOR adaptive duration.
                                                    // Set ExitAcptAmps = Same value used for LIMIT_OC_AMPS if Overcharge mode is to be
                                                    // used.
                                                    // Note: If both Time and Amps are set = 0, Acceptance will be bypassed.

    int        PH_AI;                             // Place holder. FUTURE: EXIT_ACPT_DVDT Add dV/dt exit criteria for Acceptance mode,
                                                    // need to decide what it is :-

    float      PH_AF;                             // Place holder.

                                                    // Overcharge mode is sometimes used with AGM batteries and occurs between Acceptance and
                                                    // Float phase.

    int        LIMIT_OC_AMPS;                     // During Overcharge phase, Amps are capped at this low value. (Set this = 0 to disable
                                                    // OC mode.)

    float      EXIT_OC_VOLTS;                     // Overcharge will continue until the battery voltage reaches this level.
    int        EXIT_OC_AMPS;                     // Will remain in Overcharge mode holding LIMIT_OC_VOLTS until Amps being delivered falls
                                                    // to this level or below.
                                                    // (Set = 0 to disable this exit check - and exit OC mode imidieatly upon reaching
                                                    // EXIT_OC_VOLTS)

    uint32_t   EXIT_OC_DURATION;                 // Over Charge mode duration in mS. Do not exceed this duration in total OC time. (Set =
                                                    // 0 to disable max time allowed)
                                                    // ( as a safety step, setting OC_VOLTS or DURATION = 0 will also disable OC mode..)

    int        PH_OI;                             // Place holder. FUTURE: EXIT_OC_DVDT Add dV/dt exit criteria for Overcharge mode,
                                                    // need to decide what it is :-

    float      FLOAT_BAT_V_SETPOINT;             // Set point for Float battery voltage, do not exceed this voltage.
    int        LIMIT_FLOAT_AMPS;                 // During Float, manage system to keep Amps into Battery at or under this value. May = 0,
                                                    // set = -1 to disable limit.
```

```

uint32_t  EXIT_FLOAT_DURATION;    // Alternator will stay in Float mode this int32_t (in mS) before entering Post-Float (no
                                     charging) mode. Set = 0UL disable transition to Post-float mode.
int       FLOAT_TO_BULK_AMPS;    // If Amps being delivered exceeds this value, we will assume a LARGE load has been placed
                                     on the battery and we need to re-enter
int       FLOAT_TO_BULK_AHS;    // BULK phase. Set this = 0 to disable re-entering BULK phase feature
                                     // If the number of Ahs removed from the battery after 1st entering Float mode exceed this
                                     value, revert back to BULK.
float     FLOAT_TO_BULK_VOLTS;   // Note this will ONLY be usable if the Amp shunt is at the battery. Set = 0 to disable
                                     this feature.
int       FLOAT_TO_BULK_SOC;    // As with Amps, if the voltage drops below this threshold we will revert to Bulk. Set =
                                     0 to disable.
uint32_t  EXIT_PF_DURATION;     // If the batteries SOC (as reported by a CAN attached BMS or RBM) drops below this value,
                                     reg will revert to BULK. Set = 0 to disable.

uint32_t  EXIT_PF_DURATION;    // Only stay in Post_float mode (no charging) this amount of time. Set = 0UL to disable
                                     times based Post-float exiting and exit only on Voltage.
float     PF_TO_BULK_VOLTS;    // If during Post-Float mode VBat drops below this voltage, re-enter FLOAT mode.
                                     // Set = 0.0 to disable exiting of post-float mode based on voltage.
int       PF_TO_BULK_AHS;    // Config note: IF you configure the system to enter post-float mode from float-mode (by
                                     setting a time value EXIT_FLOAT_DURATION), AND you
                                     // set both EXIT_PT_DURATION and PF_TO_BULK_VOLTS = 0, the regulator will in
                                     effect turn off the alternator once charging is completed
                                     // and not restart a charge cycle until powered down and up again. This can
                                     be useful if you truly want a one-time only charge.
                                     // You could also config the FEATURE-OUT port to indicate the complete
                                     charge cycle has finished, to say power-off the driving engine?
int       PF_TO_BULK_AHS;    // If the number of Ahs removed from the battery after 1st entering Post Float mode exceed
                                     this value, revert back to BULK.
float     EQUAL_BAT_V_SETPOINT; // Note this will ONLY be usable if the Amp shunt is at the battery. Set = 0 to disable
                                     this feature.

float     EQUAL_BAT_V_SETPOINT; // If Equalize mode is selected, this is the target voltage. Set = 0 to prevent user from
                                     entering Equalization mode.
int       LIMIT_EQUAL_AMPS;    // During equalization, system will limit Amps to this value. Set = 0 to disable amp
                                     limits during Equalization Mode.
uint32_t  EXIT_EQUAL_DURATION; // Regulator will not stay in Equalization any longer then this (in mS). If set = 0, then
                                     Equalization mode will be disabled.
int       EXIT_EQUAL_AMPS;    // If Amps fall below this value during Equalization while at VBat setpoint -- exit
                                     equalization. Set = 0 to disable exit by Amps and use only time.

float     BAT_TEMP_1C_COMP;    // Battery Temperature is compensated by this factor for every 1C temp change. Note this
                                     is based off of BAT_TEMP_NOMINAL (25c)

```

```

int      MIN_TEMP_COMP_LIMIT;      // If battery temperature falls below this value (in deg-c), limit temp compensation
                                        voltage rise to prevent overvoltage in very very cold places.
int      BAT_MIN_CHARGE_TEMP;      // If Battery is below this temp (in deg-c), stop charging and force into Float Mode to
                                        protect it from under-temperature damage.
int      BAT_MAX_CHARGE_TEMP;      // If Battery exceeds this temp (in deg-c), stop charging and force into Float Mode to
                                        protect it from over-temperature damage.

                                        // Some batteries allow extended temperature operations but require a lower max energy
                                        transfer during those times.
                                        // Note that these will NOT over-ride the BAT_MIN_CHARGE_TEMP or BAT_MAX_CHARGE_TEMP
                                        values above, but will allow pull-back as we approach them.
float    BAT_LOW_RC_VOLTS;         // If battery voltage is below this level, cap charging current to BAT_CAP_AMPS value.
                                        Set = 0.0 to disable low-volts check
int      BAT_LOW_RC_TEMP;          // If battery is below this temp (in deg-c), cap charging current to BAT_CAP_AMPS value.
                                        Set = -99 to disable lower boundary check
int      BAT_HIGH_RC_TEMP;         // If battery is above this temp (in deg-c), cap charging current to BAT_CAP_AMPS value.
                                        Set = -99 to disable upper boundary check
int      BAT_RC_AMPS;              // If one of the Reduce Charging triggers above are tripped, cap battery acceptance
                                        current to this value. (If no amp shunt, est PWM pullback) Set = 0 to disable.
                                        // Note, for safety, this is the ONE condition where the Regulator will over-ride an RBMs
                                        request battery acceptance current. Normally we follow the RBM,
                                        // but if the RBM is not configured correctly - or is just being lazy and not asking for
                                        reduced current in the case of extreme battery temps, we
                                        // we will act locally.

} tCPS;

#define BAT_TEMP_NOMINAL      25      // Nominal temp which .BAT_TEMP_1C_COMP is based around (in deg-C).
#define BAT_AMPHR_NOMINAL    500     // CPE's are based on a 500Ah 'nominal' battery size
#define BAT_VOLTS_NOMINAL    12      // which is also a '12v' battery

```

Actual content of the CPE tables. Remember, all references are against a 'normalized' 12v / 500Ah battery.

```

const tCPS PROGMEM defaultCPS[MAX_CPES] = {
//      Bulk/Accept          Overcharge          Float          Post Float          Equalization
//      Temp Comp          Extended Range
{14.1, 6.0*3600000UL, 15, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 13.4, -1, 0*3600000UL, -10, 0, 12.8, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 0,0, 0, 0*3600000UL, 0,
{14.8, 3.0*3600000UL, 5, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 13.5, -1, 0*3600000UL, -10, 0, 12.8, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 0,0, 0, 0*3600000UL, 0,
{14.6, 4.5*3600000UL, 5, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 0.005*6, -9, -45, 45, 0.0, -99, -99, 0,100}, // #1 Default (safe) profile & AGM #1 (Low Voltage AGM).
{14.7, 4.5*3600000UL, 3, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 13.2, -1, 0*3600000UL, -10, 0, 12.8, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 15.3, 25, 3.0*3600000UL, 0,
{14.1, 6.0*3600000UL, 5, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 0.005*6, -9, -45, 45, 0.0, -99, -99, 0,100}, // #2 Standard FLA (e.g. Starter Battery, small storage)
{14.4, 6.0*3600000UL, 7, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 13.4, -1, 0*3600000UL, -10, 0, 12.8, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 0,0, 0, 0*3600000UL, 0,
{14.4, 6.0*3600000UL, 15, 0,0,0, 15, 15.3, 0, 3.0*3600000UL, 0,0,0, 13.5, -1, 0*3600000UL, -10, 0, 12.8, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 0,0, 0, 0*3600000UL, 0,
{14.4, 6.0*3600000UL, 15, 0,0,0, 15, 15.3, 0, 3.0*3600000UL, 0,0,0, 0.005*6, -9, -45, 45, 0.0, -99, -99, 0,100}, // #3 HD FLA (GC, L16, larger)
{14.4, 1.0*3600000UL, 15, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 13.4, -1, 0*3600000UL, -20, 0, 12.0, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 14.4, 0, 3.0*3600000UL, 3,
0.024, -20, -20, 50, 0.0, -99, -99, 0,500}, // #4 AGM #2 (Higher Voltage AGM)
0.005*6, -9, -45, 45, 0.0, -99, -99, 0,100}, // #5 GEL
13.1, -1, 0*3600000UL, -10, 0, 12.8, 50,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 15.3, 25, 3.0*3600000UL, 0,
0.005*6, -9, -45, 45, 0.0, -99, -99, 0,100}, // #7 4-stage HD LFA (& Custom #1 changeable profile)
{14.4, 1.0*3600000UL, 15, 0,0,0, 0, 0,0, 0, 0*3600000UL, 0,0,0, 13.4, 0, 0*3600000UL, 0, -50, 12.9, 70,0,0, 0*3600000UL, 0,0, 0, 0,0,0, 0,0, 0, 0*3600000UL, 0,
0.000*6, 0, 0, 50, 0.0, 5, 45, 25, 250} // #8 LiFeP04 - Relion (& Custom #2 changeable profile)

// #1 Default (safe) profile & AGM #1 (Low Voltage AGM).
// #2 Standard FLA (e.g. Starter Battery, small storage)
// #3 HD FLA (GC, L16, larger)
// #4 AGM #2 (Higher Voltage AGM)
// #5 GEL
// #6 Firefly (Carbon Foam)
// #7 4-stage HD LFA (+ Custom #1 changeable profile)
// #8 LiFeP04 (+ Custom #2 changeable profile)
};

```

APPENDIX D: DEFAULT SYSTEM CONFIGURATION

The following documents default values (As Compiled) for the WS500 Alternator Regulator's system configuration. It is configured assuming the Amp Shunt will be placed at the battery and that a 500A / 50mV shunt is being used. (This is the shunt used in the Link-10 battery meter as well as others).

```
SCS systemConfig = {
    false,          // .REVERSED_SHUNT          --> Assume shunt is not reversed.
    100,           // .ALT_TEMP_SETPOINTc       --> Default Alternator temp - 100c
    1.00,         // .ALT_AMP_DERATE_NORMAL    --> Normal cap Alternator at 100% of demonstrated max Amp capability,
    0.75,         // .ALT_AMP_DERATE_SMALL_MODE --> Unless user has selected Small Alt Mode via DIP switch, then do 75% of its capability
    0.50,         // .ALT_AMP_DERATE_HALF_POWER --> User has shorted out the Alternator Temp NTC probe, indicating want 1/2 power mode.
    -1,          // .ALT_PULLBACK_FACTOR      --> Used to pull-back Field Drive as we move towards Idle.
    0,           // .ALT_IDLE_RPM             --> Used to pull-back Field Drive as we move towards idle.
                                Set = 0 causes RPMs to be determined automatically during operation.
    0,           // .ALT_AMPS_LIMIT           --> The regulator may OPTIONALLY be configured to limit the size of the alternator output
                                //                               Set = 0 to disable Amps capping. Set = -1 to auto-size Alternator during Ramp.
                                //                               (required Shunt on Alt, not Bat)
    0,           // .ALT_WATTS_LIMIT          --> The regulator may OPTIONALLY be configured to limit the load placed on the engine via
                                //                               the Alternator.
                                //                               Set = 0 to disable, -1 to use auto-calc based on Alternator size.
                                //                               (Required Shunt on Alt, not Bat)
    12,          // .ALTERNATOR_POLES         --> # of poles on alternator (Leece Neville 4800/4900 series are 12 pole alts)
    ((6.7 / 2.8) * 1.00), // .ENGINE_ALT_DRIVE_RATIO   --> Engine pulley diameter / alternator diameter & fine tuning calibration ratio
    (int) ((500/0.050) * 1.00), // .AMP_SHUNT_RATIO          --> Spec of amp shunt, 500A / 50mV shunt (Link10 default) and % calibrating error
                                //                               CAUTION: Do NOT exceed 80mV on the AMP Shunt input
    -1,          // .FIELD_TACH_PWM           --> If user has selected Tach Mode, use this for MIN Field PWM.
                                //                               Set = -1 to 'auto determine' the this value during RAMP phase
                                //                               Set = 0 to in effect 'disable' tach mode, independent of the DIP switch.

    true,        // .USE_BT                   --> Should we try to use the Bluetooth?
    "ALTRREG",   // .BT_NAME                   --> Name of Bluetooth module.           MAX 18 CHARS LONG! (see BT_NAME_LEN)
    "1234",      // .BT_PSWD                   --> Password to use for Bluetooth module. MAX 18 CHARS LONG! (see BT_PIN_LEN)
    DEFAULT_BT_CONFIG_CHANGED, // .BT_CONFIG_CHANGED        --> BT name and password are still the default. Updates to configuration data is
                                //                               prevented until the name & password is changed.

    0,           // .CP_INDEX_OVERRIDE        --> Use the DIP switch selected indexes
    0.0,         // .BC_MULT_OVERRIDE         --> Use the DIP switch selected multiplier
    0.0,         // .SV_OVERRIDE              --> Enable Auto System voltage detection
    0 ;          // .CONFIG_LOCKOUT;         --> No lockouts at this time.
```

```

CCS canConfig = {
    0,           // .BI_OVERRIDE           --> Battery Instance attached to. 0=use DIP switches, override with $CCN command
    1,           // .DEVICE_INSTANCE          --> Default 'Charger' instance. Override with $CCN
    70,         // .DEVICE_PRIORITY         --> Default 'Device Ranking' - 70, Below AC powered chargers. Override with $CCN command
    true,       // .CONSIDER_MASTER         --> Default, is no one else steps up to the plate - shall we try to be master? Override with $CCN
    true,       // .SHUNT_AT_BAT            --> Most installs have Shunt at battery.
    RVCDCbt_Unknown, // .BATTERY_TYPE          --> Default, we do not know unless the user tells us.
    true,       // .ENABLE_OSE              --> Default, push out OSEnergy (RV-C) messages (This is NEEDED to support remote instrumentation,
                                                                prioritization, etc.) Override with $CCN
    true,       // .ENABLE_NMEA2000        --> Default, push out NMEA-2000 messages. Override with $CCN
    false      // .ENABLE_NMEA2000_RAT    --> Default, do not look for a NMEA2000 device to remotely supply battery amperage and
                                                                temperature - and we will send out PGN: 127506
};

```

APPENDIX E: ERROR CODES AND MEANING

The following is a description of error codes as reported via the ASCII status and/or the LED blinking pattern. Most errors are hard-faults, indicating a condition which the WS500 Alternator Regulator is unable to decipher and as such will shut down until corrected, in order to prevent any potential systems or battery damage. A few errors will attempt to auto-restart to see if the failing condition clears (example, error low battery voltage).

Many error codes are related to internal logic checks, if those are received look for a firmware upgrade. However, some errors codes occur during installation errors and / or system issues. A prime example is the Alternator overheating errors – which typically indicate a need to either increase cooling and/or enable SMALL-ALT-MODE or increase its pullback. These alternator overheating issues are very common when using small frame alternators (anything under 30lbs) combined with large battery banks or any Li based battery bank.

Other error codes are related to the overall system operation. Examples are the Battery Disconnected faults, where the regulator receives notification that the BMS has (or will soon) disconnect the battery from charging sources to protect it. Any overvoltage or disconnect error must be investigated carefully to determine the cause of the overvoltage condition and corrected. (Likely miss-configuration, though perhaps incorrect auto-detect of system voltage)

FET over temperature (error #41) is an indication of a hardware issue with the regulator its self, or perhaps a short in the alternator field.

```
//---- Error codes. If there is a FAULTED status, the variable errorCode will contain one of these...
//      Note at this time, only one error code is retained. Multi-faults will only show the last one in the checking tree.
//      Errors with + 0x8000 on them will cause the regulator to re-start, others will freeze the regulator.
//      (Note combinations like 10, and 11 are not used. Because one cannot flash out 0's, and kind of hard to
//      tell if 11 is a 1+1, or a real slow 2+0)

#define FC_LOOP_BAT_TEMP          12          // Battery temp exceeded limit
#define FC_LOOP_BAT_VOLTS        13          // Battery Volts exceeded upper limit (measured via INA226)
#define FC_LOOP_BAT_LOWV         14 + 0x8000U // Battery Volts exceeded lower limit, either damaged or sensing wire missing.
(or engine not started!)

#define FC_LOOP_ALT_TEMP          21          // Alternator temp exceeded limit
#define FC_LOOP_ALT_RPMs         22          // Alternator seems to be spinning way to fast!
#define FC_LOOP_ALT2_TEMP        23          // Alternator #2 temp exceeded limit
#define FC_LOOP_ALT_TEMP_RAMP    24          // Alternator temp reached / exceeded while ramping - this can NOT be right, to
reach target while ramping means way too risky.
```

```

#define FC_LOG_ALT_STATE          31          // Global Variable chargingState has some unsupported value in check_for_faults()
#define FC_LOG_ALT_STATE1        32          // Global Variable chargingState has some unsupported value in manage_ALT()
#define FC_LOG_CPI_STATE         33          // Global Variable cpIndex has some unsupported value in caculate_ALT_Targets()
#define FC_LOG_CPINDEX           34          // Global Variable cpIndex has some unsupported value in check_for_faults()
#define FC_LOG_SYSAMPMULT        35          // Global Variable systemAmpMult has some unsupported value in check_for_faults()

#define FC_SYS_FET_TEMP           41          // Internal Field FET temperature exceed limit.
#define FC_SYS_REQUIRED_SENSOR    42          // A 'Required' sensor is missing, and we are configured to FAULT out.

#define FC_CAN_BATTERY_DISCONNECTED  51          // We have received a CAN message that the battery charging bus has been
disconnected.;
#define FC_CAN_BATTERY_HVL_DISCONNECTED 52          // We have noted that a command has been sent asking for the battery bus to be
disconnected!
#define FC_LOG_BATTINST           53          // Battery Instance number is out of range (needs to be from 1..100)
#define FC_TOO_MANY_AGGERGATION    54          // Too many different BMS's have been found in the aggregation range, internal
table size exceeded.

#define FC_ENGINE_UNSUPPORTED_INTERFACE 61          // Attempted to start and/or stop engine with unsupported 'interface'
configuration

#define FC_INA226_READ_ERROR       100 + 0x8000U // Returned I2C error code is added to this, see I2C lib for error codes.

```