VSR SYSTEMS

Communications and Programming Guide

v1.3.1 - July 22, 2018

TABLE OF CONTENTS

ABOUT THIS GUIDE

The VSR series of product offerings features a common set of configuration and communications capability. This guide is used to document communications into and out of VSR based devices (such as the VSR Alternator Regulator). Both Serial (ASCII) communications and CAN (Control Area Network) capabilities are covered; refer to the individual devices users guide for more details on which method are supported. Throughout this guide ICONs' are used to indicate which ASCII communications are supported by various devices:

⚙ = VSR Alternator Regulator

⚡ = VSR DC Generator Controller

▦ = VSR Solar MPPT controller

Note: At the time of the writing of the guide, the VSR DC Generator and VSR Solar MPPT Controller were in development stages, as such the commands noted here for these devices may be incomplete and/or change.

ASCII COMMUNICATIONS

Most all VSR Systems offering feature a built in 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.

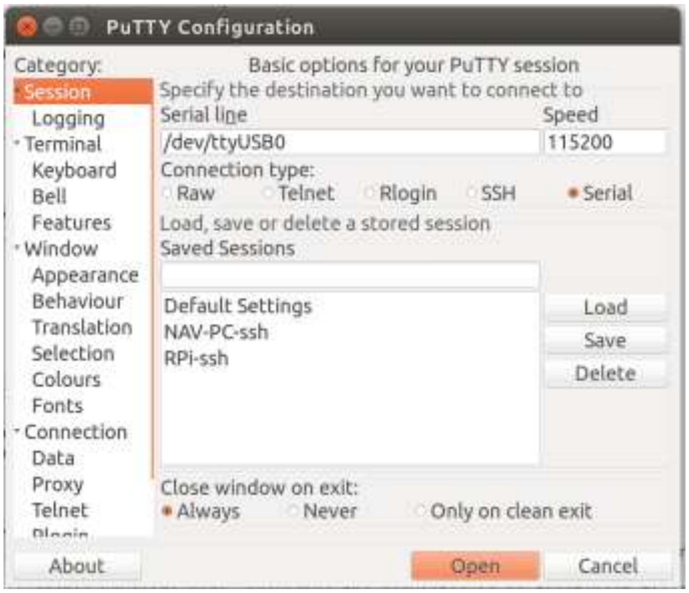
Some VSR Systems products require the use of an external USB to Serial converter – refer to the individual Reference Guides for details.

TERMINAL PROGRAMS

Many OSes have built in terminal programs, as does the Arduino IDE. After connecting the regulator to the computer you can use one of these to open a communications window to the VSR Alternator Regulator. Gen 2 regulators requires an external a USB to Serial adapter, while Gen 3 regulators already have a USB port built in device. If your Gen 2 board is equipped with the Bluetooth option, that can also be used to access the serial port on the VSR Alternator Regulator. A special note, some terminal programs do not send a complete end-of-line terminator (CR+LF - specifically, Arduino's Serial Monitor). To support these environments, the VSR Alternator Regulator will recognize the character '@' as an alternative EOL.

Remember that any configuration changes you make to the regulator via ASCII commands may not take effect until the regulator is rebooted. When you have finished, make sure to issue the $RBT: command to not only assure changes are saved to the regulators' non-volatile memory, but that the changes are then utilized by the regulator. (refer to $RBT: - ReBooT system on page 58) After rebooting the VSR Alternator Regulator verify the changes you sent were indeed recognized by the regulator by inspecting the various status strings.

Putty: Another versatile option is the free 'Putty' program (www.putty.org). It supports a wide range of OSes and includes a very nice logging function. To use connect up the USB cable as directed **Error! Reference source not found.** 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 redial button.



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

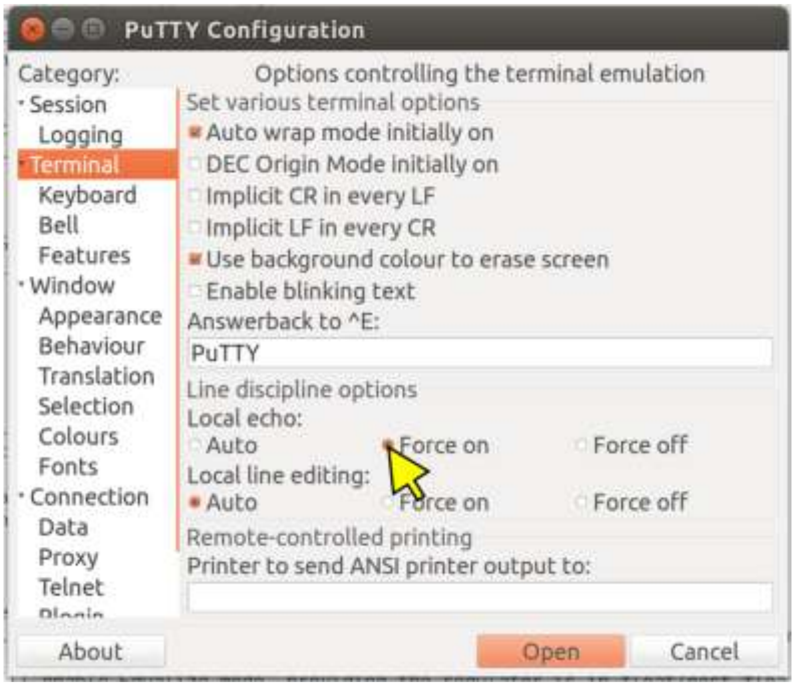
Load, save or delete a stored session

Saved Sessions

Default Settings
NAV-PC-ssh
RPi-ssh

Close window on exit: ● Always ○ Never ○ Only on clean exit

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



Options controlling the terminal emulation

Set various terminal options

☒ Auto wrap mode initially on
□ DEC Origin Mode initially on
□ Implicit CR in every LF
□ Implicit LF in every CR
☒ Use background colour to erase screen
□ Enable blinking text

Answerback to ^E: PuTTY

Line discipline options

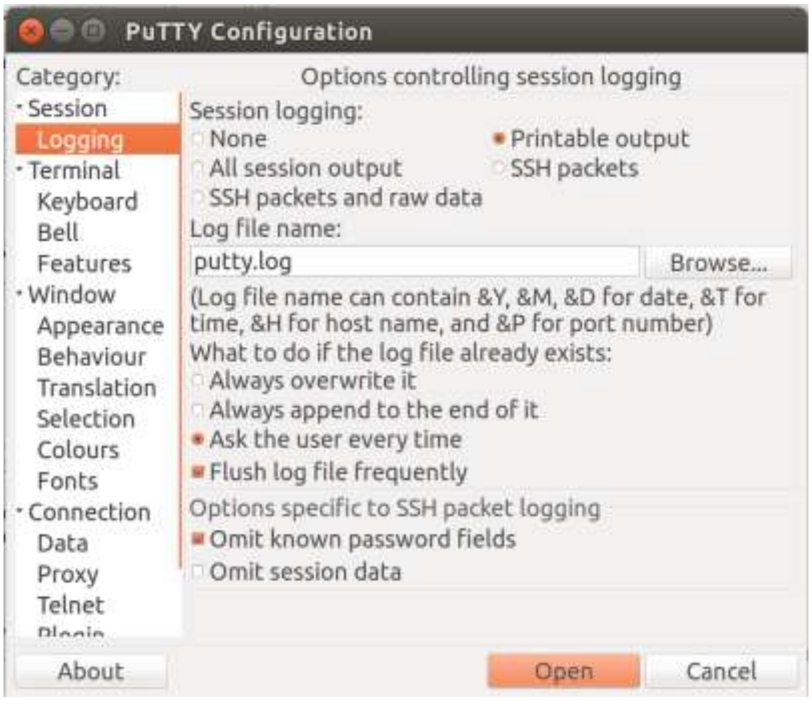
Local echo: ○ Auto ● Force on ○ Force off

Local line editing: ● Auto ○ Force on ○ Force off

Remote-controlled printing

Printer to send ANSI printer output to:

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



Options controlling session logging

Session logging:

○ None ● Printable output

○ All session output ○ SSH packets

○ SSH packets and raw data

Log file name:

putty.log

(Log file name can contain &Y, &M, &D for date, &T for time, &H for host name, and &P for port number)

What to do if the log file already exists:

○ Always overwrite it

○ Always append to the end of it

● Ask the user every time

☑ Flush log file frequently

Options specific to SSH packet logging

☑ Omit known password fields

☐ Omit session data

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 *Appendix A: Receiving data FROM the VSR* System Devices: 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 to connect the either the 2nd or 3rd generation of the VSR Alternator Regulator power is supplied to the logic portion of the hardware. This allows you to do bench-top configuration before completing the installation to the actual alternator. Make sure to do a $RBT: command as your last step, to assure changes are recorded. After the regulator reboots make sure to verify your changes before installing the regulator in a live installation.

CAN (Control Area Network) Communications

Most all VSR Systems offerings feature 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 widely used in Industrial, Heavy Industry, Farming, Medical, Consumer, and more. Over a billonCAN nodes have been deployed, with modern automobiles contained upwards of over 100 individual nodes each! It is a proven reliable and robust communications standard with many feature to assure deterministic and prioritized communication to form the backbone of the VSR Alternator Regulator.

Leveraging this backbone 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 VSR Alternator Regulator leverages several standards covering physical wiring, message content and other communications standards. These include:

- CAN Specification 2.0b / ISO-IS 11898
- CiA 303
- SAE J1939
- OSEnergy (Open Systems Energy - derived from the RV-C standard)
- NMEA2000™ (Limited support)

OSEnergy (Open Systems Energy) is an architectural specification who's 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 VSR Alternator Regulator is able deliver several key benefits, including:

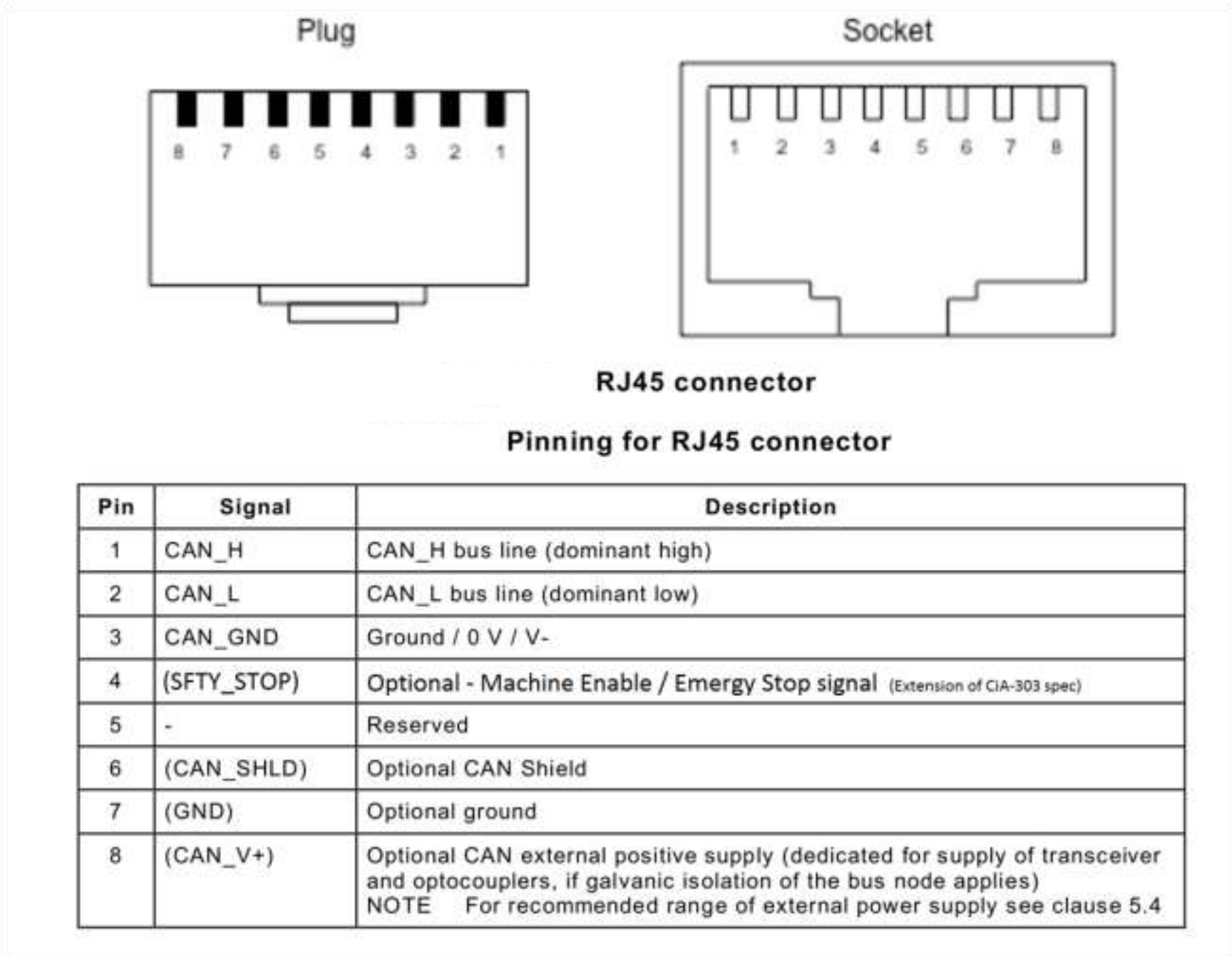
- Coordination of charging goals and objectives; all devices work towards the SAME goal vs. fighting each other.
- Prioritization of charging sources. e.g.: Utilization of Solar to its maximum capability while filling in the remaining energy needs from an engine driven alternator. Thereby saving fuel.
- Remote sensing / Port Expander: The VSR 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 LiFeP04). By using this capability wiring and installations may be simplified without reduction in reliability. (Refer to: `**Error! Reference source not found.**′ on page **Error! Bookmark not defined.** for one example)
- 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 failures, the VSR 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 leave one connector open and place a jumper on the TERM block – removing the TERM jumper on any devices in-between. 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.

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:



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 / Emergy 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-

NMEA-2000^{TM} SUPPORT

VSR Alternator Regulator CAN protocol shares the same foundation and electrical specifications and NEMA2000, and is able to produce messages which may often be displayed on a NMEA2000 device. It needs to be noted, the VSR Alternator Regulator is NOT a certified NMEA2000 compliant device, and its support is a byproduct of the J1939 library used. But in many cases connecting to a NMEA2000 network is successful. If you have issues with either the NMEA2000 or OSEnergy communications capabilities disconnect the regulator and use a stand-alone CAT-5 based network. Future product developments may result in a proper OSEnergy to NMEA2000 bridge, thereby eliminating any issues with potential conflicts.

Wiring:

To connect into an existing NMEA-2000 network you will need to make up a patch cable. The simplest way to do this is to cut one end off a common CAT-5 cable and attach a Micro connector. If you have ordered the option of a terminal strip instead of the RJ45 connectors, you may use a NMEA2000 drop cable and connect the *CAN_H* and *CAN-_L* wires to the appropriate positions. Referring to Figure 1 above as well as Figure 2 below only the *CAN_H* and *CAN-_L* wires need to be connected. With either approach take care not to exceed the maximum Drop-cable length of 6 meters, remove the TERM jumper and use the following as a wiring guide:

Signal	CAT-5 Cable			NMEA2000 Cable	
	POSITION	COLOR		POSITION	COLOR
CAN_H	1	White/Green --OR-- White/Orange		4	White
CAN_L	2	Green --OR-- Orange		5	Blue

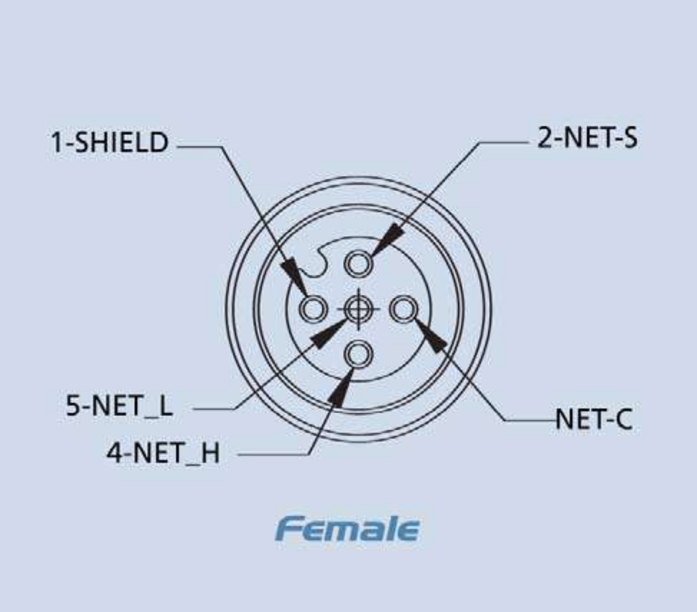


Figure 2 - NMEA2000 connector pinout

There is no need to connect the CAN-GND, and in fact doing so may cause reliability issues due to ground-loops.

A limited number of NMEA2000^{TM} like status output messages are supported. These messages may be useful when the VSR Alternator Regulator is connected to a NMEA2000 network and will allow the operational status of the alternator and battery to be displayed. Be sure to properly configure the regulator when using these messages, specifically the `Engine ID' using the $CCN: command.

- NMEA2000-DC Detailed Status - PGN127506
- NMEA2000-Battery Configuration Status -- PGN127513
- NMEA2000-Battery Status - PGN127508
- NMEA2000-Engine parameters rapid - 127488L: (Provides Engine RPMs)

The VSR Alternator Regulator is also able to receive a limited number of non OSEnergy CAN messages for special purposes:

- NMEA2000-DC Detailed Status - PGN127506
  - If `Enable_NMEA2000-RAT' is enabled (See $CCN: command), the regulator will look for DC Detailed Status to receive the battery current and temperature.
  - Eliminating the need for dedicated probes in systems which have a battery monitor already installed.
  - If the VSR Alternator Regulator receives a DC Detailed Status message, it will cease sending out one of its own.

NATIVE J1939 SUPPORT

The VSR Alternator Regulator is able to support some CAN messages beyond the OSEnergy / NMEA2000 standard. Specifically support for J1939 protocol based messages have been added for additional features and capabilities. These capabilities are often selected via compile-time switches (#defines) and require custom compiling of the source code to enable.

- J1939 Engine Speed - PGN 61444
  - A compile-option `#define DISABLE_ON_J1939_RPMs in SmartRegulator.h` when enabled will cause the regulator to go into a disabled / low power mode if Engine Speed is reported to be below 400 RPMs (indicating the engine has stopped). By enabling this compile option the ENABLE pin may be continuously powered and the regulator will look for the J1939 engine speed messages to determine if it should begin a charge cycle with the alternator or not.
  - J1939 derived RPMs will take precedence over any locally measured RPMs using the Stator wire.

Appendix A: Receiving data FROM the VSR System Devices:

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 request command (though one should still do error checking and validation, as the simple regulator will often just ignore commands that have a syntax error in them)

Formats are all 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 change 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 VSR 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 b 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:

1.2.x and below	1.3.x and above	
0,1	0,1,4	– Alternator Off
2, 3	2,3	- Alternator FAULTED (See Fault Code)
4	10	- Alternator in delay mode while engine warms up
5	11,15	- Ramping towards BULK mode.
6,7	12,20	- In BULK mode
8	21	- In ACCEPTANCE mode
9	22	- In OVER CHARGE mode
10	30	- In FLOAT mode
11	31	- In FORCED_FLOAT mode (via Feature_in pin and CPE = #8)
12	36	- In OFF (Post Float) mode
13	38	- In EQUALIZE mode
14	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 VSR 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. To use on larger systems, you will need to modify the source code and proceed at your own risk.*

GNC; -- GENERATOR CONFIGURATION

“GNC;, OP Sender, WF Sender,   ,Pre-Warm, Warm-up, Quiet RPMs”

OP Sender: PSI to indicate sufficient oil pressure in engine

WF Sender: Should the Water-flow input be sampled for faults while the generator is in Running mode?

Pre-Warm: Time in seconds the engine will be allowed to sit at idle after starting before the throttle is advanced.

Warm-up: Time in seconds during which the throttle will be slowly advanced while no load is placed on the engine.

Quiet RPMs: Max RPMs engine is allowed to run when started in Quiet Mode.

GSC; -- GENERATOR STARTER CONFIGURATION

“GSC;, Type, ,Glow, TTL Idle Adv, ,Crank, Hold off, Post Hold, Rest, Trys, , TTL Bump, Start FLD, Amps Drop, RPMs, ,Peak Amps, Cooldown”

SType: What type of starter interface does this generator have?

1. "RCI -- Run / Crank Input"
2. "CGSI -- Crank/Glow & Stop inputs
3. "PSPS -- Preheat/Start input and Prime/Stop
4. "SIO -- Single On/Off input
5. "GSCA -- Glow, Start input with Amp Shunt monitoring"

<< Paramters used in GSCA mode >

Glow: Time (in 1/10th of a second) Glow-plug is activated before engaging starter.

TTL Idle Adv: Time (in 1/10th of a second) throttle is advanced during mode before engaging starter.

Crank: Maximum time (in Seconds) starter is allowed to be engaged during a start attempt

Hold off: Time (in 1/10th of a second) sensing is held off before sampling RPM and Starter Amps to determine if engine has started.

Post Hold: Time (in 1/10th of a second) starter continues to be engaged after engine has caught.

Rest: Time (in Seconds) starter is rested between starting attempts.

Trys: Number of times the engines is attempted to be started before faulting out.

TTL Bump: Between starting attempts, throttle is advanced (bumped) this many in 1/10th of a second.

Start FLD: %drive field PWM is set when attempting to start engine

AMPS drop: %drop in measured starter amps used to indicate engine has started.

RPMs: Number of RPMs threshold to indicate engine has started.

Peak Amps: Peak amp draw of starter during last starting attempt.

Cooldown: Time engine is to spend at idle w/o load before stopping.

GTC; -- GENERATOR THROTTLE CONFIGURATION

“GTC;, Warmup bump, Bump, Settle,   ,Idle RPMs, Max RPMs, Engine HP,   ,Alt Eff%”

Warmup bump: Duration throttle control motor is advanced while engine is warming up.

Bump: Duration throttle control motor is adjusted while engine is operating

Settle: Duration throttle control motor is held unchanged between making adjustments.

Idle RPMs: RPMs the engine is specified to idle at.

Max RPMs: Maximum continuous RPMs the engine is specified for.

Engine HP: Maximum continuous HP engine is specified for.

Alt Eff%:   % Rated efficiency of attached alternator to generator.

CST; -- CAN STATUS

"CST;, BatteryID, IDOverride, Instance, Priority, ,Enable NMEA2000?, Enable OSE?, ,AllowRBM?,IsRBM, ShuntAtBat?,
,RBM ID, IgnoringRBM?,Enable_NMEA2000-RAT?, ,CAN_ID, ,EngineID"

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 type 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 VSR 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 VSR 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 = VSR 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_NMEA2000-RAT?: 0 or 1, Allow NMEA-2000 device to Remotely supply battery Amperage and Temperature information via the Battery Status PGN: 127506. If *enable_NMEA2000-RAT* is set = yes, the regulator will not send out its own copy of PGN: 127506 (1 = Yes, default = 0)

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

EngineID: Engine ID (or number) the VSR Alternator Regulator with the engine it is mounted on. Used for monitoring J1939 RPMs messages as well as sending NMEA2000 RPMs back out.
(New in V1.1.1 firmware)

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' vales. See Defining Charging Voltages and Amps for additional information.

"CPE;, n, acptVBAT, acptTIME, acptEXIT, res1, , ocAMPS, ocTIME, ocVBAT, res2, , floatVBAT, floatAMPS, floatTIME, floatRESUMEA, floatRESUMEAH , floatRESUMEV, ,pfTIME, pfRESUME, pfRESUMEAH, , equalVBAT, equalAMPS, equalTIME, equalEXIT, , BatComp, CompMin, MinCharge, MaxCharge"

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

res2: Reserved for future use (dV/dT exit criteria, currently = 0, disabled)

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.

GST; -- GENERATOR STATUS

"GST;, Engine State, ,ETemp, EGT, SWT, EMT, ,Oil PSI, PreFilter PSI, HP PSI, , Aux Engaged, , Total Hours"

Engine State: Current state of the generators Engine, per the following table:

0,1,4	– Generator is OFF or SLEEPING.
2,3	- Generator FAULTED (See Fault Code)
50,51	– Generator is STOPPED or Stopping
52	– Generator is STARTING
53, 54	– Generator is WARMING UP
55	– Generator is RUNNING
59	– Generator is doing special prime-oil function.

ETemp: Measured engine temperature of NTC sensor attached to Engine Temp port in degrees C. -99 indicates temperature has not been measured, NTC sender has failed, or is not attached.

EGT: Exhaust Gas Temperature in degrees C. -99 indicates EGT probe is not attached, or has failed.

SWT: For installations which use a water-to-water heat exchanger, this is the temperature of the NTC probe installed on the raw cooling water (or sea water) pump in degrees C. -99 indicates SWTemp has not been measured, NTC probe is not attached, or has failed.

EMT: Exhaust Mixer Temperature in degrees C. For systems which utilize a 'wet' exhaust system this temperature sensors indicates the temperature of the Exhaust Mixer in degrees C. -99 indicates EMTemp has not been measured, NTC probe is not attached, or has failed.

Oil PSI: Engine Oil pressure in PSI. -99 indicates oil pressure sensor is not installed or has failed. 999 indicates oil sender is simple on/off switch and is indicating oil pressure is present.

PreFilter PSI: For installations which contain an auxiliary load, specifically a Water Maker, this is the Pre-Filter pressure (or vacuum) in PSI. -99 indicates PreFilter sensor is not installed or has failed.

HP PSI: For installations which contain an auxiliary load, example Water Maker or diving compressor, this is the PSI of that load. -99 indicates sensor is not installed or has failed.

Aux Engaged: For installations which contain an auxiliary load, this indicated the status of the engagement clutch. 1=Engaged, 0=disabled.

Total Hours: Number of hours engine has been in RUNNING state for lifetime of controller.

MST; -- MPPT STATUS

MST: “MST;, Hours, ,MPPT Volts, BatVolts, MPPT Amps, BatAmps, ,TargetVolts, TargetAmps, TargetWatts, ,MPPT State, Panel Volts, Panel Amps, ,BatTemp”

CAUTION: This is a PRELIM spec for the MPPT controller, largely to show the potential for other offerings.

- - This is HIGHLY likely to change - -

Hours: Time MPPT has been in this charge cycle up, in hours and fraction (to 2 digits) of hours.

MPPT Volts: Measured Volts output of MPPT controller, in volts and fractions of volts (to 1mV resolution)

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

MPPT Amps: 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 VSR 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 b 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.

MPPT State: Current state of the Alternator, per the following table:

State	

39	- In CVCC mode (only available in system under direction of CAN master)

AltVolts: Measured Alternator Volts, in volts and fractions of volts (to 1mV resolution)

Panel Volts: Measured Panel Volts, in volts and fractions of volts (to 1mV resolution)

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

Bat Temp: 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.

*note: * If the VSR MPPT Controller is configured with no limits for Amps and/or System Watts a self-impose limits of 1,000A / 15,000W as max values. MST; will report these working values*

NPC; -- NAME & PASSWORD CONFIGURATION

"NPC;, Use BT?, Name, Password, , SerialNum”

use BT? 0 or 1: Enable Bluetooth? (1 = yes)

Name: Name of Regulator (Used for Bluetooth and CAN device ID) (ASCII up to 18 characters)

Password: Password (Used for Bluetooth PIN) (ASCII up to 18 characters)

SerialNum Serial number of VSR Alternator Regulator (Not supported in all generations)

Note: This status string was called BTC; in versions of firmware before v1.0.0

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"

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

Reserved: Always = 0; *(Was Favor 32v system detection over 24/48v?)*

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.

(Firmware 1.2.0 and above)

Required Sensor: Key indicaing critical sensors which have been configured via the $SCA: command
(Firmware 1.3.0 and above)

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, RequriedSensorStatus ”

System has Faulted, fault code number (See Source code for details.)

RequiredSensorStatus is a combined number following the pattern as defiend by the $SCA command and detailed in `Table 1 - Required Sensor Encoding ‘ on page number 47

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. See Source code for details of Debug String

RST; -- RESET

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

APPENDIX B: SENDING DATA TO THE REGULATOR:

All commands begin with the character `$', contain 3 letters(CAPS) followed by a `:' and then parameters as requested by the command. All must end with a CR (or CR/LF) or may optional be terminated with the character `@' (this is to accommodate a bug in the Arduino IDE that does not send CR nor LF at the end of entered strings). A complete 'string' must be received within 60 seconds from the `$' to the ending `@'/CR/LF, else the regulator will abort the capture of that string command and begin looking for a new `$' starting character. (See *#define IB_BUFF_FILL_TIMEOUT 60* in source code).

Sending communications TO the regulator:

Remember: On Version 2 (Bluetooth capable) until you initially the regulator's NAME and PASSWORD from their default values via the $SCN command, no ASCII other change command will be processed. See 'Error! Reference source not found.' on page# Error! Bookmark not defined.

Defining Charging Voltages and Amps – All volts and Amps are represented for 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 (and Bluetooth if enabled). 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 advanced setup parameters of the VSR Alternator Regulator which are not supported via standard RV-C DGNs. See AltReg_CAN.ccp for more details.

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

This command (with its parameters) 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> Place holder for future t.b.d. dV/dT exit criteria. 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 OVERCHARGE 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>, <Reserved>

---

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 OVERCHAGE phase, delivered Amps will be monitored and regulated to *'LimitAmps'* . Setting *'LimitAmps'* = 0 will disable OVERCHAGE mode.

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

Exit VBat: <FLOATING POINT NUMBER (0.0 → 20.0) > Once battery voltage reached *'ExitVBat'*, OVERCHAGE 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 OVERCHAGE mode.

Reserved: <0> Place holder for future t.b.d. dV/dt exit criteria. Must be 0.

$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>

---

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: <WHOLE 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.

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>
```

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 phase. Setting ‘*ExitDuration’* = 0 will disable POST-FLOW mode revering to FLOAT charge immediately.

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

$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 → 240 (4 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^{O}C>, < Min Comp Temp >, <Min Charge Temp>, <Max Charge Temp>

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 (-30 → 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, the system will be forced into FLOAT mode to protect it.

Max Charge Temp: <WHOLE NUMBER (20 → 95) > If the battery reaches this temperature, the system will be forced into FLOAT mode to protect it. If the battery temperature continues to raise, the system may eventually FAULT based on the value of #define FAULT_BAT_TEMP in the source code. 60c (140f) by default.

$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.

$CGN: - Configure GeNerator

This command is used to adjust the Generators Configuration.   Parameters must be in the following order and include comma “,” separators where indicated. Extra spaces before and/or after the parameters are allowed.

$CGN: <OP Sender>,<WF Sender>, <Pre-Warm>, <Warm-up>, <Quiet RPMs>

OP Sender: <WHOLE NUMBER (-2 → 100) > What is the minimum oil PSI the engine should be allowed to operate it? If oil-pressure is above this value before attempting to start the controller will FAULT as it appers the engine is already running. If while running the Oil Pressure drops below this value the controller will FAULT and stop the engine.

If set = 0: Oil-press checks will not be performed during starting nor while engine is operating.

If Set = -1: Checks will be made, but it will be assumed the attached sender is an on/off sender vs. a pressure sender; with Off (open) indicating a lack of oil pressure (N.O. type sender).

If Set = -2: Checks will be made, but it will be assumed the attached sender is an on/off sender vs. a pressure sender; with Off (open) indicating sufficient oil pressure (N.C. type sender).

WF Sender: <WHOLE NUMBER (0 → 1) > Should the Water-flow input be sampled for faults while the generator is in Running mode? (0=No, 1=Yes)

Pre-Warm: <WHOLE NUMBER (0→ 300) >   Time in seconds the engine will be allowed to sit at idle after starting before the throttle is advanced.

Warm-up: <WHOLE NUMBER (0→ 300) >   Time in seconds during which the throttle will be slowly advanced while no load is placed on the engine. After warm-up time the status will change to Running and the generator load will be applied.

Quiet RPMs: <WHOLE NUMBER (200→ 10,000) >   If the generator is stated requesting Quiet Mode, RPMs will be capped to this maximum value (vs the defined Max RPMs of the engine). The generator (alternator) load will be reduced to match the capability of the engine at this RPM. Set to a value equal to or higher (suggested 10,000) to disable Quite-mode capability of generator.

$CGS: - Configuration Generators Starter

This command is used to adjust the Generator's Starter Configuration. Parameters must be in the following order and include comma "," separators where indicated. Extra spaces before and/or after the parameters are allowed.

$CGS: <Type>,<Glow>,<TTL Idle Adv>,<Crank>,<Hold off>,<Post Hold>,<Rest>,<Tries>,<TTL Bump>,<Start FLD%>,<Amps Drop%>,<RPMs><Cool down>"

Type: <WHOLE NUMBER (1 → 5) > Indicates method DC Generator controller is to use for communicating to the engine for starting and stopping. Depending on the configuration selected, additional parameters which follow may (or may not) be utilized. Make sure to always provide a complete configuration strings, even if some parameters are unused by a given start/stop interface method – it is suggested to send in the DISABLE values where appropriate. *'Type'* is defined as one of the following methods:

Method	Start/Stop Interface
1	RCI -- Run / Crank Input
2	CGSI -- Crank/Glow & Stop inputs
3	PSPS -- Preheat/Start input and Prime/Stop
4	SIO -- Single On/Off input
5 (Default)	GSCA -- Glow, Start input with Amp Shunt monitoring

Refer to the *`VSR DC Generator Reference Manual'* for more details on each of these engine start/stop interface methods.

Glow: <WHOLE NUMBER (0→ 600) > Number of seconds the Glow-Plug should be enabled before starting is attempted. Set = 0 to disable Glow-Plug

TTL Idle Advance: <WHOLE NUMBER (0→ 100) > Time in 1/10ths of a second the throttle control motor is advanced before starting is attempted. (100 = 10.0 seconds) Adjust this value to match your engines configuration so that the throttle will be advanced to a typical start/idle position. Set =0 to disable throttle advance before starting is attempted.

Crank: <WHOLE NUMBER (0→ 300) > Maximum time in second the starter will be engaged during each attempt to start the engine. If engine does not catch, or 'fire' during a starting attempt – the starter will be disengaged after *'Crank'* seconds and rested before another starting attempt is made. Set = 0 to disable engagement of starter during starting attempts.

Hold off: <WHOLE NUMBER (0→ 50) > Time in 1/10ths of a second the VSR DC Generator Controller will wait before beginning to monitor RPMs and Starter Amperage while determining if the engine has started. Primarily used to avoid sampling of the starters initial surge current. Set = 0 is starter current is not being used to as one method to determine if engine has fired.

Post Hold: <WHOLE NUMBER (0→ 50) > Time in 1/10ths of a second the starter remains engaged after the VSR DC Generator Controller has determined the engine has caught/fired. Set = 0 to immediately disengage the starter after an indication of starting has been determined.

Rest: <WHOLE NUMBER (0→ 300) > Number of seconds the starter is allowed to rest between starting attempts.

Tries: <WHOLE NUMBER (1→ 10) > Number of times the VSR DC Controller should attempt to start the engine. If the engine is not determined to have started after *'tries'* number of start-cycles the generator will be placed in a FAULTED state.

TTL Bump: <WHOLE NUMBER (0→ 20) > Time in 1/10ths of a second the throttle motor will be advanced between starting attempts. Set = 0 to disable any throttle advancement between tries.

Start Field %: <WHOLE NUMBER (0→ 100) > % of field drive that should be applied to the alternator's Field while attempting to start the engine. Useful if RPMs is being used as one of the ways to determine if the engine has fired/caught. After the engine is started the Field drive will be set to its minimum value. Set `Start Field%' = 0 to disable any field excitement while starting is attempted.

Amps Drop %:<WHOLE NUMBER (0→ 100) > One of two methods used by "GSCA" to determining if the engines has fired during a starting attempt. This method will monitor the starter amperage draw with a reduction of at least *'Amp Drop%'*, vs. the steady-state starter current draw will be an indicator the engine has caught/fired (Starter amperage reduced due to load being removed from it as engine starts to run).

Set = 0 to disable starter amperage current monitoring for a method for indicating engine has started.

RPMs: <WHOLE NUMBER (0→ 2000) > 2nd method used by "GSCA" to determine if engine has started. If engine RPMs exceeds this value, it will be assumed engine has started. Set = 0 to disable engine RPM method for determine if engine has started.

> *CAUTION: If the engine interface method GSCA is selected make sure to enable at least starter Amperage drop % or engine RPMs as a way to determine if engine as started. Disabling both approaches will result in the VSR DC Generator controller continue to engage the starter – even if the engine is already started.*

Cooldown: <WHOLE NUMBER (0→ 1000) > Time in Seconds engine should remain at idle with no load before finally stopping.

$CGT: - Configure Generators Throttle

This command is used to adjust how the Generator's Throttle is managed. Parameters must be in the following order and include comma "," separators where indicated. Extra spaces before and/or after the parameters are allowed.

$CGT: <Warmup bump>,< Bump >,<Settle>,<Idle RPMs>,<Max RPMs>,<Engine HP>,<Alt Eff%>

Warmup bump: <WHOLE NUMBER (0→ 100) > Time in 1/10ths of a second the throttle control motor is advanced while engine is warming up. The '*Warmup Period'* (See $CGN) is divided into 20x divisions. At each 1/20th time segment the throttle is advanced this amount. Set =0 to disable throttle advance during engine warm-up period.

Bump: <WHOLE NUMBER (0→ 100) > Time in 1/10ths of a second the throttle control motor is advanced while engine is warming up. The '*Warmup Period'* (See $CGN) is divided into 20x divisions. At each 1/20th time segment the throttle is advanced this amount. Set =0 to disable throttle adjustments while engine is operating.

Settle: <WHOLE NUMBER (0→ 100) > Time in 1/10ths of a second the throttle control motor is held unchanged after making and adjustment before another adjustment is attempted (if needed). Allows the engine to stabilize before making additional adjustments. . Set =0 to disable

Idle RPMs: <WHOLE NUMBER (200→ 2000) > RPMs the engine is specified to idle at. Used to defile slowest speed the VSR DC Generator Controller will operate the engine at under a load, and also used to define the lower-point of the engine RPM/HP calculations.

Max RPMs: <WHOLE NUMBER (500→ 10000) > Maximum *continuous* RPMs the engine is specified for.

Engine HP: <WHOLE NUMBER (500→ 10000) > Maximum *continuous* HP engine is specified for.

Alt Eff%: <WHOLE NUMBER (20→ 100) > Rated efficiency of attached alternator to generator.

Asdf //!! HEY!! Talk about how these four values are used to define a load-curve for the the engine, and alternator is used to match that.

is specified for.

to idle at. Used to defile slowest speed the VSR DC Generator Controller will operate the engine at under a load, and also used to define the lower-point of the engine RPM/HP calculations.

$RGC: - Restore Generator Configuration to default values

Restores to default (values at compile time) all Generator Configuration values ($G*C:) to their default values.

```
$RGC:
```

$RAS: - Request All Status back

This command will instruct the VSR 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 paced 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 VSR 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 VSR Alternator Regulator is rebooted it will not utilize those values. For current active targets being used, look at AST; and SCV; status strings.

N**ote on Change Requests to Charge Profiles:** The source code currently will allow ONLY Charge Profile 7 or 8 (the two customizable entries) to be modified via an ASCII command. This is to reduce the potential for major errors in the regulator. If you wish to modify other Charge Profiles, the Source code will need to be revised to either change the default tables, or alter the trap to allow changes via ASCII beyond entry 7 or 8.

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 VSR Alternator Regulator may be deployed with the Amp Shunt wither on the Alternator, or the battery, there 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.

$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>, <Alt Amp Cap >, <System Watt Cap. >, <Amp Shunt Ratio>, <Shunt Reversed?>,<Idle RPMs>,<Warmup Delay>,<RequiredSensors>

Reserved: <WHOLE NUMBER = 0>   Was 'Favor 32v?', a feature which was been removed due to potential risk for false auto-selection.   Send '0' for this now reserved parameter.

Alt Target Temp: <WHOLE NUMBER (15 → 120)> Operating temperature the regulator should attempt to keep the Alternator under. If the Alternator temperature exceeds this value, the regulator will reduce field current to allow the alternator to cool off. if the Alternator temperature continues to raise and exceeds this temperature by 10% (as defined by #define FAULT_ALT_TEMP) the regulator will fault out and stop all power production.

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 VSR 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 bring 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 VSR 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 temperate 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 reliability 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.

Alt Amp Cap: <WHOLE NUMBER (-1→ 500) > This 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.

Note: ***Alt Amp Cap*** is a feature intended to be used in Alternator-Centric deployments (see section '*Example Installations'* ***Error! Reference source not found.***) and **should *ONLY* be defined if the current shunt is located *at the alternator.*** During operation the regulator will limit measured Amps to *Alt Amp Cap* value. During reduced power modes (i.e., – Half Power mode) the Amperage allowed will be further limited by the appropriate scaling factor.

For Battery-Centric deployments this value should be set = 0, thereby disabling measured amperage limits of the alternator. With Amperage limits 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 VSR 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 increases 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 = 60 seconds.

(Available with Firmware 1.2.0 and above)

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 then 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
Current Shunt	4	FAULT regulator (See note**)
Engine Temperature Sensor	8	Go into Falf Power Mode, stop Watermaker
EGT Temperature Sensor	16	Go into Half Power mode, Stop Watermaker, Full throttle.
Sea-water(cooling) Temperature Sensor	32	Fault if missing
Watermaker PSI (pre / post) Sensors	64	Disable Watermaker
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 if the Feature_out port is configured to drive a dash-lamp (compile time default mode) it will turn on the lamp full time indicating a fault.

The VSR 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.

Note** It is difficult to determine if an Amp Shunt has failed vs. if are truly reading 0A of current. Because of this, the VSR Alternator Regulator will delay check for the presence of a working Current Shunt until after Bulk has been completed. If at any time during BULK a current of greater than 5A was noted it will be flagged as the shunt is present and working. Once this determination is made no additional checks will be made – as a valid operation condition for the regulator is a true 0A of current (example, when actively regulating current to 0A in FLOAT mode).

(Available with Firmware 1.3.0 and above)

$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
  - Be sure to unlock 2nd gen regulators first, see "**Error! Reference source not found.**" on page **Error! Bookmark not defined.**
- 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 D:)*

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

Notes: Some versions of the Arduino IDE do not send the correct cr/lf termination – in these cases the '@' symbol (as shown above) may be placed at the end of a line to communicate end-of-command.

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@
$CPP:7 0,0,0@
$CPE:7 15.3,0,180,0@
$CPB:7 0.030,15,-50,125@
$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@
```

$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 VSR 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 VSR 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 then 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 wit 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 you battery.

If a fixed *Tech 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 *Tech 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 (2..4) and Battery Capacity (5..6) 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.

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. (If auto SV feature is overridden the favor32v flag will have no impact.) 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 then 0), it can ***ONLY*** be cleared by doing a hardware based master reset (See Feature-In, mode 1 above), 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. See section "Restore to AS-Compiled (default) Status"

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

- If not already done, set regulators name (Needed to unlock regulator, see "**Error! Reference source not found.**" on page **Error! Bookmark not defined.**)
- 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, and allows forced disable of Bluetooth (if equipped). The name is used by the optional Bluetooth module as well as the CAN controller 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:   <Enable BT?>, < Reg Name >, <Reg Password>
```

Enable BT?: <WHOLE NUMBER (0, or 1)> Should the Bluetooth be enabled (in conjunction with DIP-Switch 1, Bluetooth power)? Use this to turn off Bluetooth via software.

0 = Disable Bluetooth until master reset or enabled via Service port. ,
1 = RE-enable Bluetooth adapter (providing DIP Switch is turned on).

Caution: If you disable the Bluetooth via software, the only way you will be able to re-enable it is via attaching a physical TTL serial cable to the Service / USB Port, or by doing a master restore on the regulator.

Reg Name: <STRING (up to 18 characters, no spaces, comma, or '@') > Name used for Bluetooth broadcast as well as 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 '@') > Password that should be asked for when an external device is attempting to attach to the Bluetooth.

Comment on Bluetooth Security: Bluetooth was design to simplify communications between personal devices in close proximity to each other. Part of this capability is easy visibility and connection. This however has a downside of the risk of someone coming close to your alternator and 'hi-jacking' the alternator via Bluetooth; even opening up the potential for malicious activity that can damage your system. The Regulator has several security features to help prevent this:

1. The ability to change Charge Parameters or System Configuration is disabled until the factory default name and password are changed.

2. The name and password can only be initially changed if the Alternator is not charging, and all the DIP switches are ON.
3. When attaching and external computer or tablet, you must enter the password in your host.
4. The password can be long (up to 18 character), and alphanumeric.
5. Once you are happy with the configuration, there is the ability to lock-out future changes using the $SCO command.

The password is your primary line of defense, please choose it carefully, make it random, and long. Do not use 1234 or 0000 (Common ones). If you have great concern, utilize the $SCO lockout feature.

$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 (+ Bluetooth) to default

Restores System Configuration AND Bluetooth values to original as-compiled (default).

$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

(This command is only recognized by the 3rd generation CAN enabled regulator)

Configure the CAN Configuration Table for this regulator

$CCN: <Battery Instance Override>, <Device Instance >, <Device Priority>, <AllowRMB?>, <ShuntAtBat?>, <Enable-OSE?>, <Enable-NMEA2000?>, <Enable_NMEA2000_RAT?>, <Engine ID>

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. (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). (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 VSR 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 VSR 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(default), 1=yes

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

Enable NMEA2000_RAT?: <WHOLE NUMBER (0, or 1)> Should the VSR Alternator Regulator look for a NMEA2000 device to supply remotely sensing battery Amperage and Temperature via PGN: 127506? 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: 127506. 0=no, 1=yes(default)

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.
(New in v1.1.1 firmware)

$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

(This command is only recognized by the 3rd generation CAN enabled regulator)

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. It performs the same function as holding FEATURE-IN high during initial power-on (See *"RESTORE TO AS-COMPILED (DEFAULT) STATUS"*)). 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. (Serial + Bluetooth). This has the like effect to adding *"#define DEBUG"* to the source code, except $EDB: will ONLY enable the serial strings. Other debug features (ala, overriding Bluetooth lock outs, etc) will not be changed. This command is also only effective for the time the Regulator is running. 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 stronigly 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 regulators', 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.

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 meet, the regulator will again exit that phase quickly. In such cases the mode may be chanced before the next $AST string is sent.

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 C: CAN MESSAGES

Beginning with the 3rd generation of the VSR Alternator Regulator a built-in Control Area Network (CAN) is included. The purpose of this network is to allow communications of status, configuration, and coordination of charging in a systems view.

The VSR 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. Closes specification built upon J1939 which has been reverse engineered.
- RV-C -- (RV-C.com) True open source speciation targeting primary Recreation Vehicle industry, but extended to include many needed communications to support the OSEnergy initiative.

CAN messages summary

The following summarizes the pgns sent out vie the VSR Alternator Regulator over the CAN bus. This list may change; refer to the source code for more details.

'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 heath
        //  - 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 Ã‚Â°K. Use function CToKelvin, if you want to use Ã‚Â°C.
```

```
//  - SID                   Sequence ID.
*/

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

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 Tempeature     -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
*/

/*****************************************************************************
// 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
*/

////////////////   THIS IS A PROPOSED ONE!!!!!!  ???????????????????????????????
/*****************************************************************************
// Charger Status2 - 1FF9Dh  (PROPOSED, TEMP USING OLD BRIDGE_DGN_LIST DGN #)
// 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            -2M..+2MA, in 1mA steps (0x77359400 = 0A)
//  - Temperature           -40..210 in deg-C, in 1C steps
*/


/*****************************************************************************
// Charger Configuration Status - 1FFC6h
// Input:
```

```
//  - Instance
//  - Charging Algorithum
//  - 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
*/
```

The following table indicates how often CAN messages are sent. As an example pgn: 127506 (NMEA-2000 DC Status message) is sent every 667mS, while pgn: 127513 (NMEA-2000 Battery Configuration Message) is not automatically sent out, but will only be sent when a request is made (via the J1939 ISO-Request message pgn: 59904). One is also able to see which PGNs are utilized by the VSR Alternator Regulator via the _handler entries.

This table may change as features are added, refer to the source code for the most recent update.

```
    tCANHandlers CANHandlers[]={
        #ifdef SUPPORT_NMEA2000                                      //  Sent | N2K| RVC | RBM   <<-- Characterization flags, when are these messages
allowed, have they been sent recently, etc..
        {127506L,NULL,                   &N2kDCStatus_message,      667,{false,true,false,false}},   // 0x1F212
        {127508L,&N2kDCBatStatus_handler,&N2kDCBatStatus_message,  667,{false,true,false,false}},   // 0x1F214
        {127513L,NULL,                   &N2kBatConf_message,         0,{false,true,false,false}},   // 0x1F219
        {127488L,NULL,                   &N2KEngineParamRapid,      500,{false,true,false,false}},   // 0x1F200 // Note I am using 500ms vs. 100mS per N2K
Spec.  (Why SOO fast?)
        #endif
        #ifdef SUPPORT_RVC
        {0x1FFFD,&RVCDCStatus1_handler, &RVCDCStatus1_message,     500,{false,false,true,true}},
        {0x1FFFD, NULL,                 &RVCDCStatus1OA_message,   100,{false,false,true,true}},
        {0x1FFFC,&RVCDCStatus2_handler, &RVCDCStatus2_message,     500,{false,false,true,true}},
        {0x1FEC9,&RVCDCStatus4_handler, &RVCDCStatus4_message,    5000,{false,false,true,true}},
        {0x1FEC8,&RVCDCStatus5_handler, &RVCDCStatus5_message,     500,{false,false,true,true}},
        {0x1FEC8, NULL,                 &RVCDCStatus5OV_message,   100,{false,false,true,true}},           // Special instance, called every 100mS when
over voltage.
        {0x1FEC7,&RVCDCStatus6_handler,           NULL,              0,{false,false,true,false}},
        {0x1FED0,&RVCDCDisconnectStatus_handler,  NULL,              0,{false,false,true,false}},
        {0x1FECF,&RVCDCDisconnectCommand_handler, NULL,              0,{false,false,true,false}},
        {0x1FFC7,&RVCChrgStat_handler,  &RVCChrgStat_message,     5000,{false,false,true,false}},
        {0x1FF9D,&RVCChrgStat2_handler, &RVCChrgStat2_message,     500,{false,false,true,false}},            /*  PROPOSED!!!  USING TEMP PGN# */
        {0x1FFC6,&sendNAK_handler,      &RVCChrgConfig_message,      0,{false,false,true,false}},            // For now we do not allow CPE configuration
via the RVC protocol.
        {0x1FF96,&sendNAK_handler,      &RVCChrgConfig2_message,     0,{false,false,true,false}},
        {0x1FECC,&sendNAK_handler,      &RVCChrgConfig3_message,     0,{false,false,true,false}},
        {0x1FEBF,&sendNAK_handler,      &RVCChrgConfig4_message,     0,{false,false,true,false}},
        {0x1FF99, NULL,                 &RVCChrgEqualStat_message,5000,{false,false,true,false}},
        {0x1FF98,&sendNAK_handler,      &RVCChrgEqualConfig_message, 0,{false,false,true,false}},
        {0xFEEB,  NULL,                 &RVCProdId_message,          0,{false,false,true,false}},
        {0x17E00,&RVCTerminal_handler,  &RVCTerminal_message,       50,{false,false,true,false}},            // Terminal handler called every 50mS to send
out 'Next portion' of string.
        #endif
        #ifdef SUPPORT_J1939
        {0xF004,&J1939EngineSpeed_handler,         NULL,              0,{false,false,true,false}},
        #endif

            // Common J1939 type messages we need to handle.
        {0x1FECA, NULL,                 &ISODiagnostics_message,  5000,{false,false,true,false}},
        {0x1FECA, NULL,                 &ISODiagnosticsER_message,1000,{false,false,true,false}},            // Special instance, sent out more often
during fault condition.
```

```
        {0,NULL,NULL,0,false}                                                                              // ----PGN of 0 indicates end of table----
        };       // Note:  send_CAN() scans this table from the beginning each time looking for the next message who has timed out and is ready to be 
sent.
       //        As such, this table becomes a priority order for CAN messages.  be careful placing a very low time period message early in the table
       //        as that may cause that one message to dominate transmissions - preventing later entries from being serviced.
```

Table 2: CAN messages sent and received

APPENDIX D: DETAILS OF CPE (CHARGE PROFILE ENTRIES)

The following are excerpt from the CPE.H source code file to give more details on how each parameter impacts battery charging.

```
               //----- This structure defines a 'profile' for battery charging.  Each stage consist of 'modes', primarily:  Bulk, Acceptance,
               //      Overcharge, and Float.  Each mode has a max voltage set point, and criteria for exiting that phase (Exceeding a time limit,
               //      or Amps dropping below a given value).   Of special note is the entry Float and Post Float, which have additional criteria
               //      resuming charging.
               //
#define MAX_CPES        8                                        //  There are 8 different Charge profile Entries
#define CUSTOM_CPES     2                                        //  The last two of which are set aside as 'customizable' and are changeable via the ASCII
string commands.

typedef struct {                                                 // Charging Profile Structure

   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.
   unsigned long  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.
                                                                 //
                                                                 // FUTURE:  EXIT_ACPT_DVDT  Add dV/dt exit criteria for Acceptance mode, need to decide
                                                                    what it is :-)



                                                                 // 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.
   unsigned long  EXIT_OC_DURATION;                              // Over Charge mode duration in mS.
                                                                 // ( as a safety step, setting OC_VOLTS or DURATION = 0 will also disable OC mode..)
                                                                 // 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.  Maybe =
                                                                0, set = -1 to disable limit.
unsigned long  EXIT_FLOAT_DURATION;                          // Alternator will stay in Float mode this long (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
                                                             // BULK phase.   Set this = 0 to disable re-entering BULK phase feature
int            FLOAT_TO_BULK_AHS;                            // If the number of Ahs removed from the battery after 1st entering Float mode exceed this
                                                                value, revert back to BULK.
                                                             // Note this will ONLY be usable if the Amp shunt is at the battery.  Set = 0 to disable
                                                                this feature.
float          FLOAT_TO_BULK_VOLTS;                          // As with Amps, if the voltage drops below this threshold we will revert to Bulk.  Set =
                                                                0 to disable.



unsigned long  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.
                                                             // 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.
                                                             // 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.
unsigned long 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, then 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.

   } CPS;
```

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

```
const tCPS PROGMEM defaultCPS[MAX_CPES] = {

       //      Bulk/Accpt                 Overcharge                              Float                            Post Float              Equalization                  Temp Comp

       {14.1, 6.0*3600000UL, 15,        0,  0.0,   0*3600000UL,        13.4,  -1, 0*3600000UL, -10,  0, 12.8,    0*3600000UL, 0.0, 0,     0.0,  0,    0*3600000UL, 0,     0.004*6, -9, -45, 45},
       {14.8, 3.0*3600000UL,  5,        0,  0.0,   0*3600000UL,        13.5,  -1, 0*3600000UL, -10,  0, 12.8,    0*3600000UL, 0.0, 0,     0.0,  0,    0*3600000UL, 0,     0.005*6, -9, -45, 45},
       {14.6, 4.5*3600000UL,  5,        0,  0.0,   0*3600000UL,        13.2,  -1, 0*3600000UL, -10,  0, 12.8,    0*3600000UL, 0.0, 0,    15.3,  25, 3.0*3600000UL, 0,     0.005*6, -9, -45, 45},
       {14.7, 4.5*3600000UL,  3,        0,  0.0,   0*3600000UL,        13.4,  -1, 0*3600000UL, -10,  0, 12.8,    0*3600000UL, 0.0, 0,     0.0,  0,    0*3600000UL, 0,     0.004*6, -9, -45, 45},
       {14.1, 6.0*3600000UL,  5,        0,  0.0,   0*3600000UL,        13.5,  -1, 0*3600000UL, -10,  0, 12.8,    0*3600000UL, 0.0, 0,     0.0,  0,    0*3600000UL, 0,     0.005*6, -9, -45, 45},
       {14.4, 6.0*3600000UL,  7,        0,  0.0,   0*3600000UL,        13.4,  -1, 0*3600000UL, -20,  0, 12.0,    0*3600000UL, 0.0, 0,    14.4,  0,  3.0*3600000UL, 3,     0.024,  -20, -20, 50},},
       {14.4, 6.0*3600000UL, 15,       15, 15.3, 3.0*3600000UL,        13.1,  -1, 0*3600000UL, -10,  0, 12.8,    0*3600000UL, 0.0, 0,    15.3,  25, 3.0*3600000UL, 0,     0.005*6, -9, -45, 45},
       {13.8, 1.0*3600000UL, 15,        0,  0.0,   0*3600000UL,        13.36,  0, 0*3600000UL,   0,-50, 12.9,    0*3600000UL, 0.0, 0,     0.0,  0,    0*3600000UL, 0,     0.000*6,  0,   0, 40}
       };
```

```
const CPS PROGMEM defaultCPS[MAX_CPES] = {
               // #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 E: Default System Configuration

The following documents default values (As Compiled) for the VSR 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.
        90,                   // .ALT_TEMP_SETPOINTc          --> Default Alternator temp - 90c  (Approx 195f)
        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?
        "ALTREG",             // .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
        false,                // .SHUNT_AT_BAT           --> Until user explicitly tells us otherwise, we need to assume the shunt is NOT connected to the
                                                             battery when we are the RBM.
        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
    };
```

Note: not all struct entries shown here, see Source code for more details.

APPENDIX F: 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 VSR 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).

These error codes may change and be expanded, refer to source code for latest list. 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. Prime examples are the Battery Disconnected faults, where the regulator received 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 #40) 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_REQIRED_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_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.
```