ALTERNATOR REGULATOR

AND OPEN-SOURCED INTELLIGENT ALTERNATOR REGULATOR

v1.0.1 - Jan 28, 2017
For use with Source Code v1.0.1 and later

Acknowledgments

Special thank you to those who assisted debugging things, correcting code, documents, and many other ways of help. The Alternator Regulator is better thanks to their contribution.

Antti-Pekka Virjonen

Terry Slattery

TABLE OF CONTENTS

The Smart Alternator Regulator

"House Batteries" are common on RVs, boats, remote cabin / houses. These battery banks often represent investments in the hundreds if not $1,000s of dollars, so it is well worth the effort to take care of them.

The use of an external alternator regulator is well known way to improve over standard 'auto' internal regulators. With accurate measurement of battery voltage, and then control of the alternators field to adjusts it output, these often provide several important features such as soft staring to save belts, multi-stage charging to assure faster and more complete recharging, even temperature sampling of the battery to adjust target voltages as needed. But with almost universal exception, all currently available external chargers are lacking one key ability: the ability to monitor current as well as voltage.

Sampling the specific gravity of the Acid is the often preferred way to determine the true SOC of a battery. Another approach recommended by many battery manufactured during recharging is to monitor the amount of current a battery is accepting[1]. By continuing to hold a battery in the Acceptance Phase until these manufactures recommended thresholds are meet we can assure a battery is indeed fully recharged. Without the ability to measure current most common regulators revert to another approach for determining if a battery is fully recharged: they guess.

The Smart Alternator Regulator includes the ability to monitor current in addition to voltage and temperatures. It also provides the ability to limit alternator output to protect the alternator, battery and/or engine (depending on how it is configured). And it is able to support batteries and alternators from 12v to 48v, and both P type as well as N type alternators.

During installation an Amp Shunt is placed on the house battery. This allows access to monitoring current into and out of the battery. You may already have such a shunt already installed, for example of your use a battery monitor along the lines of the Link-10. In this case, simple reuse the same amp shunt, though do make sure the regulator is correctly configured (default is 500A / 50mV) with the shunt proprieties – refer to the $SCA: command for more details.

Another way to utilize the Smart Alternator Regulator is to place the amp Shunt on the Alternator, as opposed to locating it at the battery. With this configuration, the Amps monitored can more closely focus on the Alternator and one is able to configure the Regulator to limit the amount of Amps produced, and hence the load the alternator places on the engine. This can be useful in cases of building a DC generator where the Alternator is much larger than the engine is able to support – by capping the amount of power the Alternator is allowed to produce you can match its demands to the capability of the engine. A downside to this approach is one looses true visibility into the House battery state of charge; other loads on the system can cause confusion – ala, if there is a 'house load' to power instruments, navigating equipments, and such, this confuses slightly the true status of the batteries state of charge. One can compensate for this by either adjusting the Charge Profiles, slightly raising the Amp Exit Thresholds to account for an expected average house-amp draw, or the ASCII command $EOA: can be used by an external all-charging source cordoning device to inform the Smart Alternator Regulator of any adjustments to the measured Amps that are appropriate.

---

[1] Example, see: "Exide Battery Charging & Storage Guidelines 5_9_13" -- www.exide.com/Media/files/Downloads/TransAmer/Battery Care and Maintenance/Battery Charging & Storage Guidelines 05_9_13.pdf

The Smart Alternator Regulator may also be used in conventional Voltage Only mode[2] by simply not connecting up the Amp Shunt (Suggested to place a small wire across the Amp Shunt terminals to remove the chance of any electrical noise fooling the regulator). The Regulator will fall back to time only charge profiles, though with very accurate measurement of voltages. In addition, Acceptance Phase will utilize an 'adaptive' time based formula, it will remain in Acceptance for 5x the duration the regulator was in Bulk mode, OR the a configured maximum amount of time contained in the CPE – whichever is less. In this way the battery gains more protection from over charging when the regulator is unable to measure the amps.

[2] Voltage Only mode is detected when the Arduino Alternator Regulator is unable to measure current in excess of 5A at any point in time. Under this case the code will ASSUME the Amp Shunt is either not connected, or damaged and will fall back to time-only exit criteria. (ala, all the CPE Amp values will be assumed set =0 – Disabled). See source code "#define USE_AMPS_THRESHOLD" to control this capability.

THE SMART ALTERNATOR REGULATOR FAMILY

There are three members in the Smart Alternator Regulator family. All share common characteristics as noted above, including: flexible 12v-48v support, ability to use acceptance amps to properly charge a battery, and tight voltage regulation. But there are some slight differences between each version:

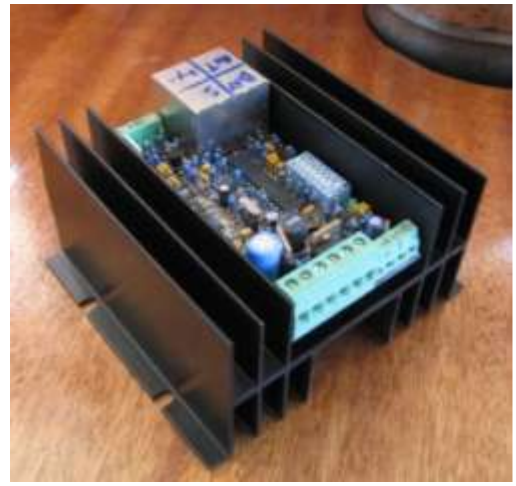

Figure 1 - First Generation Design

1st Generation: (RETIRED) Original development design. Based on the ATmega328 CPU, PCB version 0.0.x - this design is retired and not supported.

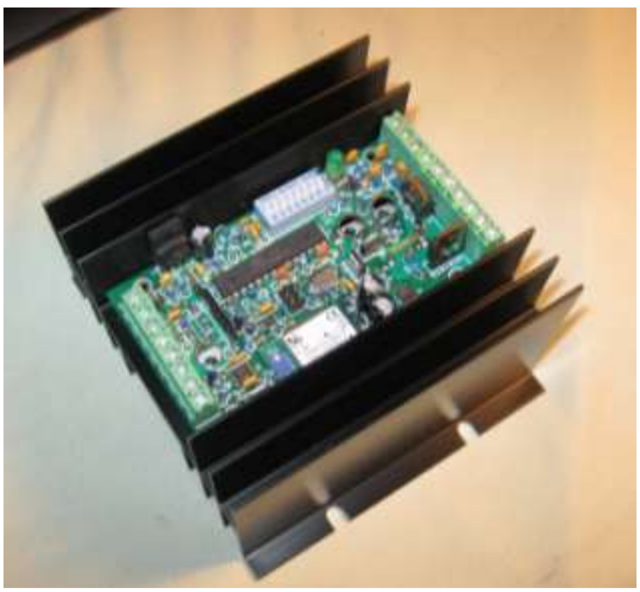

Figure 2 - Second Generation Design

2nd Generation: (ACTIVE) Through-Hold components. Continuing to use the ATmega328P CPU, PCB versions 0.1.x. Features optional Bluetooth. Made available as blank PCB - v0.1.4 being the most common.

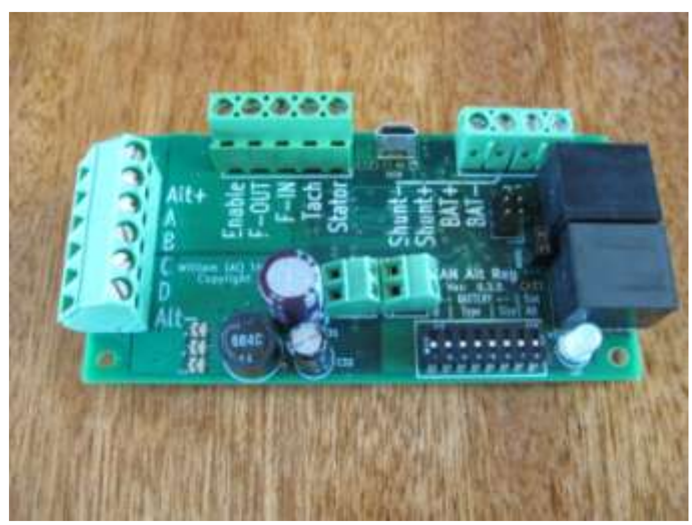

Figure 3 - Third Generation Design

3rd Generation: (ACTIVE) Newest version - SMT based design. Utilizes the ATmega64M1 CPU,   PCB version 0.3.x it features CAN subsystem. Available as both blank PCBs as well as partially/fully assembled regulator beginning with v0.3.5 of the PCB

Development and support for both the 2nd and 3rd generation design are active. The major difference being the 2nd generation features an optional Bluetooth module, while the 3rd generation includes CAN (Control Area network) ports to allow for status updates, as well as forming the basis for coordination and cooperation of multiple charging sources (See **Error! Reference source not found.** on page#**Error! Bookmark not defined.**). Both utilize the same source code – board specific selection are automatically made during compile time via #IF statements in the source.

Notable Changes API changes with Source Code v1.0.1 (released Jan 28, 2017)

With the v1.0.0 release of the source code there were a few changes to the ASCII status strings and ASCII command strings. Generally these were additions to existing strings, or new strings. However some existing details have been changed, notably:

- $AST; -- Added VAlt, FLD%, Alt2Temp & FETTemp,
- $SCV; -- Added Idle RPMs value, and TechFieldMin.
- $SCA -- Manage idle RPMs configuration now uses % field drive vs. raw PWM values.
- $SCV; & $SCA now reference Favor_32v vs. favor_36v
- $SCB ASCII command replaced by $SCN
- BTC Status string replaced by NPC status string
- $SST: -- Version number moved to beginning of string.

Reference `*Appendix A: Receiving data FROM the regulator:'* and `*Appendix B: Sending data TO the regulator:'* for details on these changes as well as new commands.

Of special note is the change of favor 36v to favor 32v. If you have a 32v or 36v battery you will want to pay special attention to this change, and also read carefully the section entitled: `*Special Considerations for 32v/36v and 42v systems'* on page 8

SMART ALTERNATOR REGULATOR CONNECTIONS

The following illustrates connection terminals on the Smart Alternator Regulator. See the following table for a description of each connection, as well as suggested min size. Example deployments follow the table.

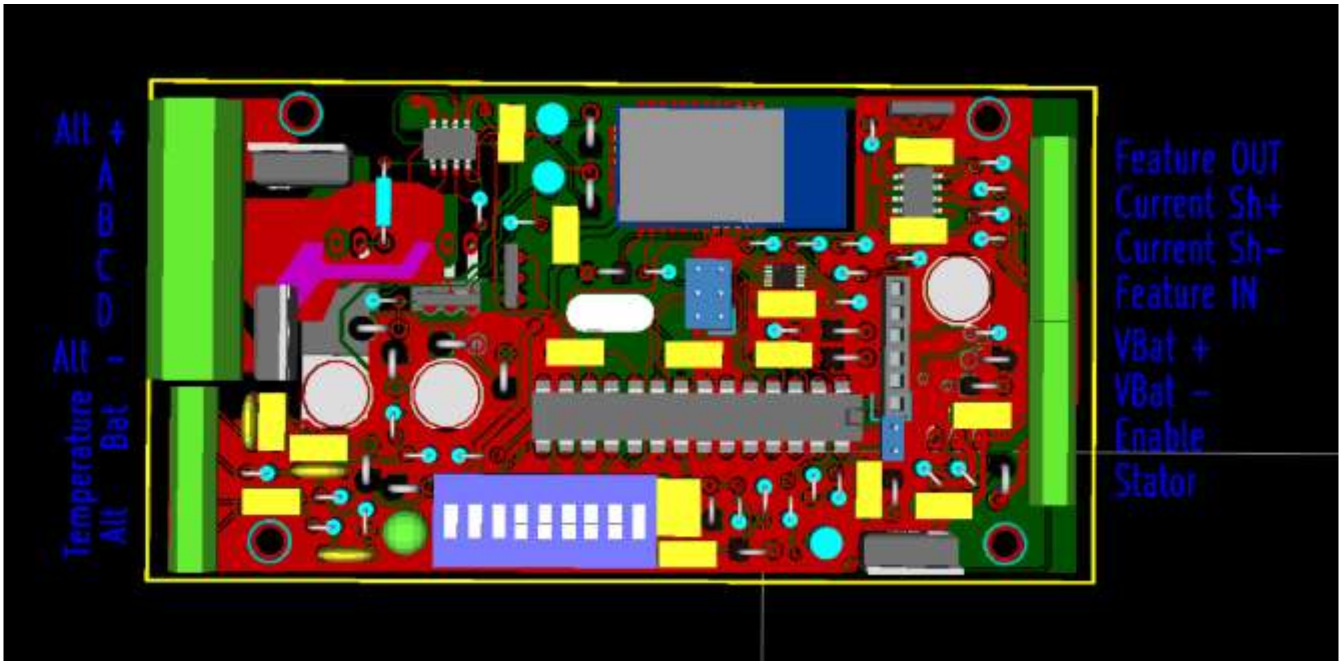

Figure 4: 2nd Generation connection

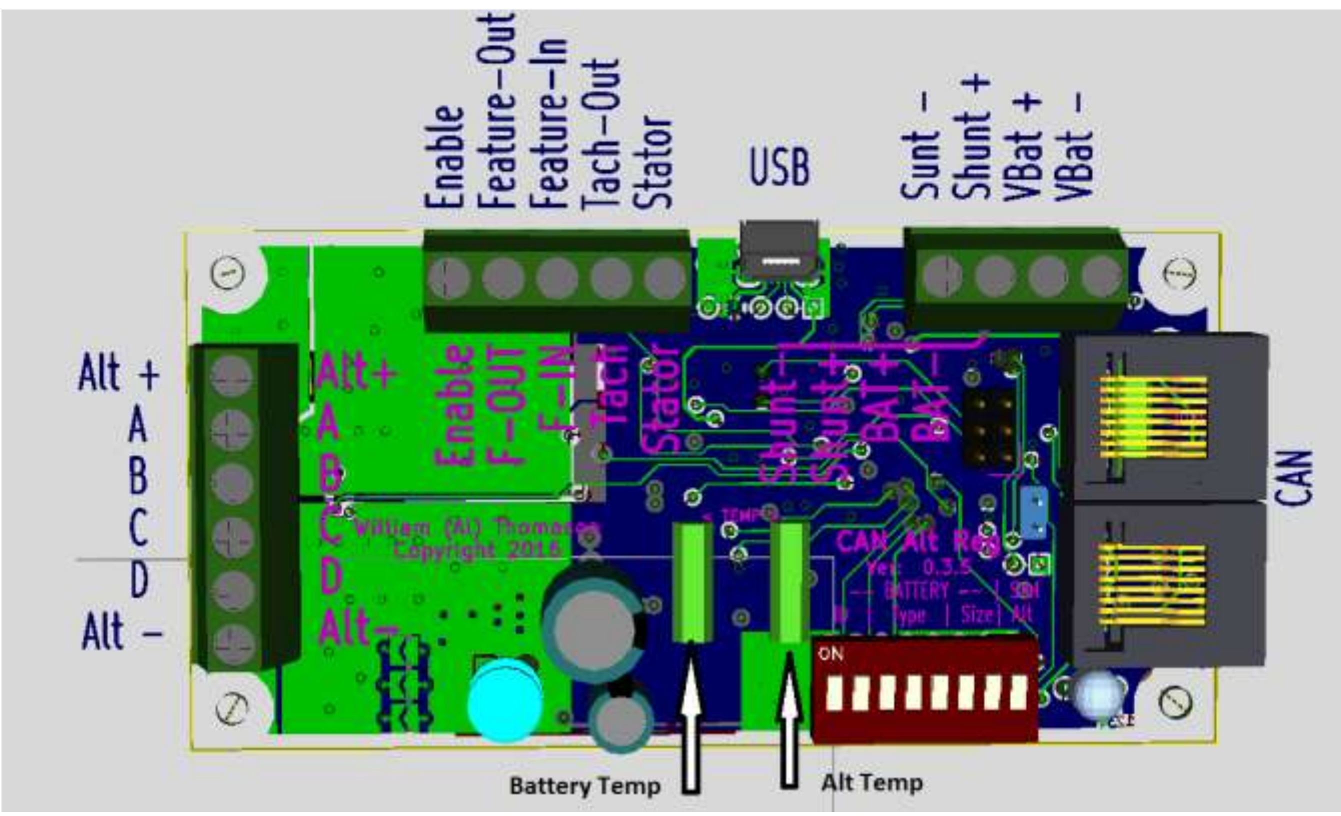

Figure 5: 3rd Generation connections

VBat+, VBat - Connect ***directly*** to the system battery via 14g wire protected with a 2A fuse.
(Do not connect after any busses, shunts, etc..)

Alternatively these may be connected locally to the alternator if the regulator will remotely receive battery voltage via the CAN bus. Refer to "Systems Installation" in Figure 9 on page 14

Do not exceed 72 volts.

Enable: Connect to VBat+ to 'turn on' the regulator. Use min 14g wire and a 2A fuse.
Do not exceed 72 volts.

Current Shunt + ,
Current Shunt -: (optional) Connect to the Current Shunt using twisted pair 16 or larger wire. The Current Shunt maybe installed in either a ground wire (low shunt), or in the + voltage wire (High Shunt). Do not exceed 80mV difference between CS+ and CS-, nor exceed connect to a shunt more than 72v above ground. If the Current Shunt is not being used, it is suggested to place a wire between these two terminals to avoid any electrical noise confusing the regulator.

Feature In: (optional) Connect to VBat (6-72v) to enable features (see FEATURE IN section)

Feature Out: (optional) Open Collector driver , connect to external Alternator LAMP at dash. 0.5A *max* current sink capability. See Source Code to enable other optional capabilities.

Alternator +: Connect to + (Bat) terminal of Alternator. Use wire sized to match your expected maximum Field current draw and protected by an appropriate fuse. (Min 14g)

Alternator -: Connect to the – (gnd) terminal of Alternator using appropriate wire. (Min 14g)

Stator (optional) Connect to an Alternator Stator pole via 2A fuse and 16g wire.
Used to increase battery voltage measurement accuracy, as well as enable several battery and alternator protection features in the regulator.

A, B, C, D: Connect to the field per the following table depending on the configuration of your alternator:

	Jumper	**Alternator Field**
High Drive (P / B-type)	**A - B**	**Field: C**
Low Drive (N / A-type)	**C - D**	**Field: A**

Use wire of sufficient gauge to carry the expected current (Min 14g).

Bat Temp,
Alt Temp: (Optional) Appropriate NTC temperature sender.
Note that Alt Temp may be OPTIONAL shorted to enable half-power mode.

Service / USB: Used to initialize and debug the regulator. Some version contain built in USB connector, others require use of external USB ← → TTL adapter.

CAN: (Some versions) Allows communication of regulators status via NMEA-2000 and/or OSEnergy protocols. Provides for remote sensing of battery and charger coronation / prioritization with other OSEnergy compliant devices. Utilize CAT-5 cables if regulator is populated with RJ-45 connectors, otherwise use 120 Ohm twisted pair wire to the CAN terminal block.

Tach: (Some versions) A conditioned signal to help drive alternator sourced tachometers – even at low charging levels.

EXAMPLE INSTALLATIONS

The following will give an overview of how to connect and configure the Smart Alternator Regulator in different situations. There are two distinct ways the Smart Alternator Regulator can be configured and installed in a system, depending on where the Amp Shunt is placed. If the shunt is placed at the battery, the regulator is able to accurately monitor the SOC (State of Charge) of the battery and use that to determine when it should change charging phases. (ala, from Acceptance to Float). This is the default deployment model for the regulator.

Alternatively, the amp shunt can be placed at the Alternator. In this deployment model the regulator can be configured to limit the amount of Amps / Watts the Alternator produces. This is useful to either further protect a smaller alternator, or perhaps to allow a much too large alternator to be placed on a small engine (ala, a DC Generator).

SPECIAL CONSIDERATIONS FOR 32V/36V AND 42V SYSTEMS

The Smart Alternator Regulator self-adjusts for battery voltage, applying an appropriate multiplier to the CPE entries. For example, if deployed with a 24v battery all the CPE voltage values will be doubled. VBat Target might go from 14.4v to 28.8v. The multiplier factor being used is reported out in the $SST ASCII status string.

In Auto-select mode (default) the Smart Alternator Regulator samples battery voltage at each startup and based on the voltage measured will decide what the most likely system battery voltage is. There is however some cross over between the high end of a 24v system, and the low end of a 32v system, as well as the high end of 32v systems and 48v. So that in a 24v system if there is another charging source active, the system voltage could be over 30v. The Smart Alternator Regulator then has to decide if this is a 24v system with another charging source active, or a 32v system with a really depleted battery. By default, it will guess a 24v system – not only because 24v systems are more common, but if it had selected 32v when the system was indeed 24v, some damage can occur. So the safe bet is to favor 24v systems. However, if you have a 32v system you can tell the regulator to favor 32v when faced with this ambiguity via the $SCA command. Perhaps a better / more reliable option for 32v systems is to override automatic selection and force the battery voltage multiplier to 2.667x via the $SCO command.

To deploy in a 32v system you can use the $SCO command to fix the voltage multiplier at 2.667. This will adjust all CPE entries to match the needs of your 32v battery. After doing so, make sure to review all other configuration voltages and make appropriate changes. Likewise, as 42/48v batteries start to make an appearance (driven largely by the Auto industry) a multiplier of 3.5 / 4.0 (respectively) may be used.

EXAMPLE 1: BATTERY CENTRIC INSTALLS (DEFAULT)

In this example the driving engine is much larger than any load the alternator will place on it, and we want to focus on the needs of the battery – specifically we want to sense when the battery is indeed fully charged before changing out of Acceptance phase. In order to do this the Amp Shunt is placed at the battery, an example connection diagram:

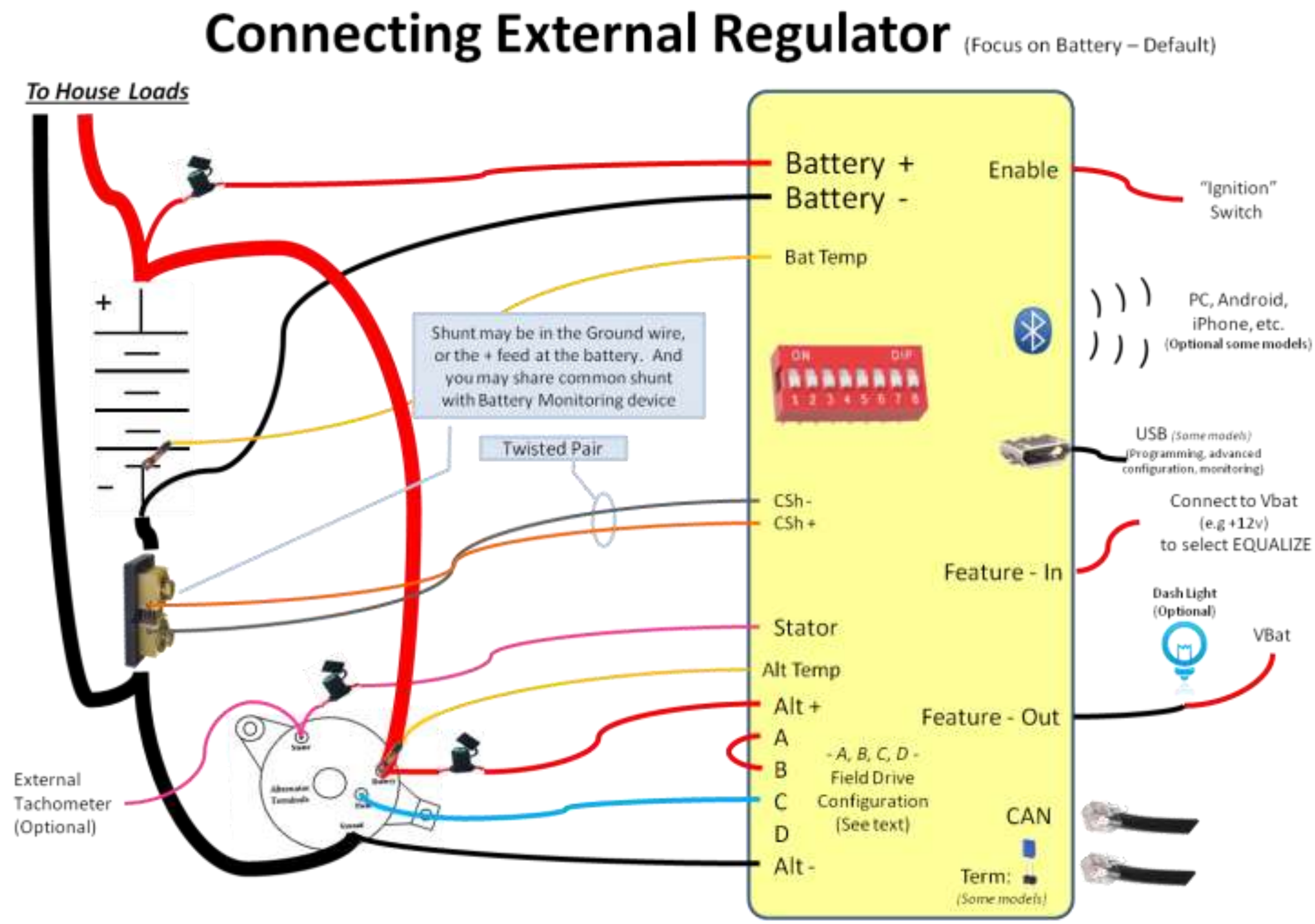

Figure 6: Wiring Diagram (Battery Priority)

By locating the amp shunt directly at the battery the Smart Alternator Regulator is able to account for any other charging sources, as well as potential house loads – to give a true indication of the status of the battery's state of charge. A side note, the shunt can actually be shared with other functions, such as an Amp meter, another alternator regulator. It can be placed on either the Ground side of the battery (as shown), or on the + side. If you have an existing battery monitor system – there is no need to install a 2nd shunt, just use the one already in place (though you may need to adjust the system configuration using the $SCV command if the shunt is not a 500A/50mV shunt).

When installing the battery voltage sensing wires (Battery+ and Battery -) make sure they go DIRECTLY to the battery! Do not attach the wires after a battery switch, dual alternator diode separator, or the Battery Amp Shunt, or a common 'buss bar'. Instead connect directly to the batteries for best results.

Some special notes on deployments:

Twin Engine Installations: It is common for many marine applications to have two engines, each with an alternator to charge the batteries. In this case, simply install a regulator on each engine and connect as shown. You can share the same battery Current Shunt between both regulators, and in this way both will know what is happening. Configure the two regulators the same and connect the Enable wire to each respective engine. It is best if each regulator hs its own Battery + and Battery – sensing wires, and the temperature sensors ***cannot*** be shared, each will need their own.

It would be a good idea to use a different Bluetooth name for each of these regulators, perhaps "Great Boat Port" and "Great Boat Star"

Diode based battery isolators: These are often installed when one alternator is asked to charge two or more batteries. the Isolator prevents any loads on one battery from discharging the other battery when the engine is turned off. These present a problem for the Smart Alternator Regulator, as it is attempting to decide when the battery is full – which battery does it look at? If you have a battery Diode battery isolator, it is might be better to replace it with a automatic battery switch (see next). However, if you do install the Smart Alternator Regulator in a a system with a Diode battery isolator, it is suggested you pick the battery you with to focus on – place the Battery Amp Shunt on that battery, and attach the battery voltage sensing wires to that same battery. The regulator will then control the Alternator to meet the needs of THAT battery. There is a risk of overcharging the 2nd battery, but that risk existed well before installing the Smart Alternator Regulator…

Automatic Battery Combiner: Another way to connect a 2nd battery to the main one is to use an Automatic Battery Switch. These will sense the voltage of both batteries, and when the time is right connect them together. If you install in a system with one of these, again connect the Smart Alternator Regulator to the primary battery, placing the Battery Shunt on that main battery and connecting the Battery + and Battery – sensing wires to that battery. The Smart Alternator Regulator will focus on that main battery and let the Automatic Battery Combiner deal with the needs of the 2nd battery. A couple of notes:

- Make sure to connect the 2nd battery on the Alternator side of the Battery Amp Shunt – it is important that the Amy Shunt ONLY measure the current needs of the battery we are focusing on.
- There is a #define FEATURE_OUT_COMBINER option in the source code that can be enabled to allow a simple external high current relay to be used for a battery combiner. See the section on Feature-out options.

EXAMPLE 2: USING REGULATOR WITH A SMALL DC GENERATOR (ADVANCED)

Another way to deploy the Smart Alternator Regulator is to focus more on the Engine and Alternator (though not totally ignoring the battery). An example would be a DC generator where the Alternator is so large it can easily stall the engine; we need to reign it in some. By placing the Amp Shunt in the Alternator feed, we are able to watch closely the output of the Alternator and prevent an overload condition.

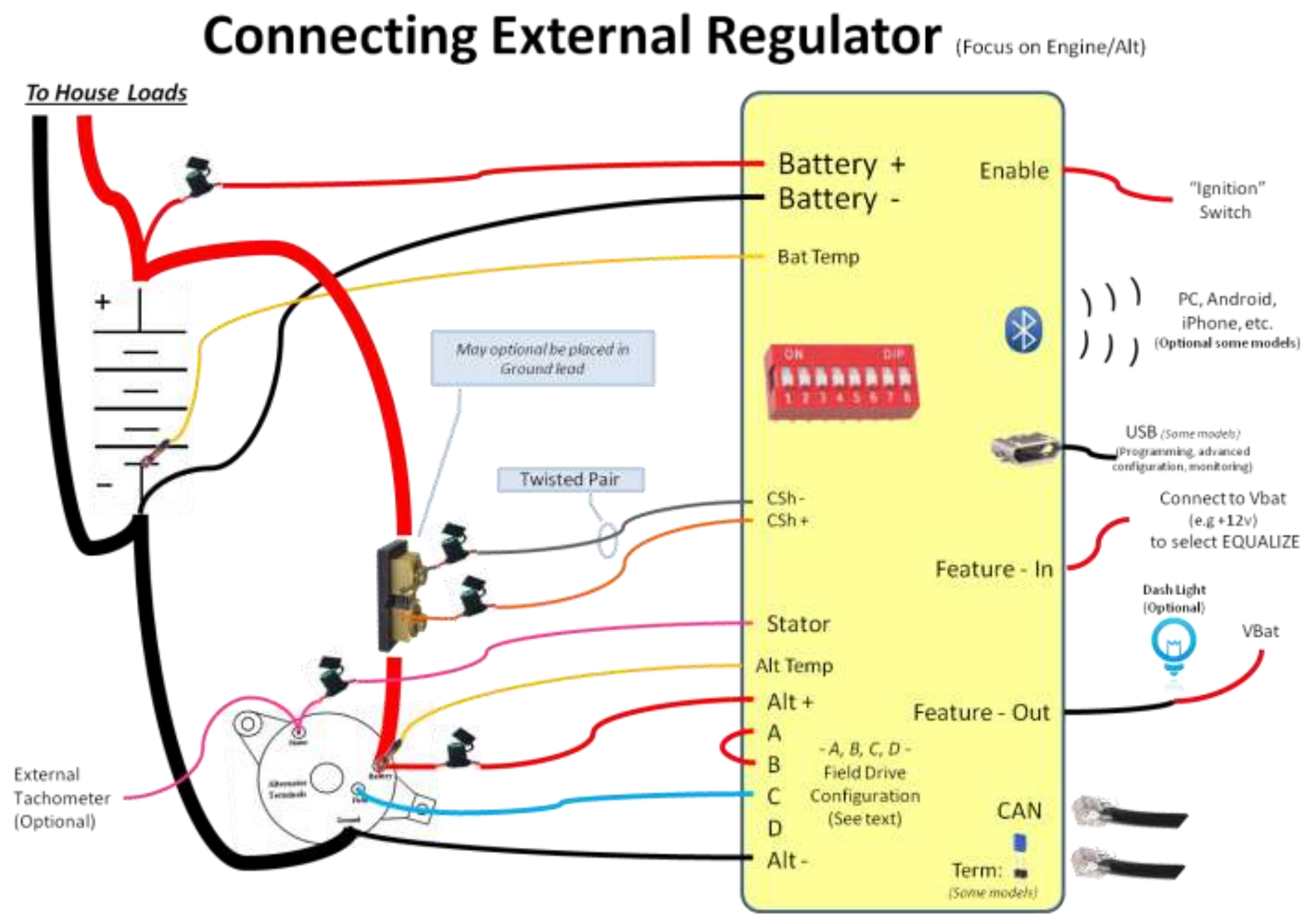

Figure 7: Wiring Diagram (Engine/Alt Priority)

By monitoring the number of amps being produced by the alternator, and configure the regulator (via the $SCV command) to the maximum number of Watts we wish to be placed on the engine the max engine load can be controlled. You may also want to adjust other parameters, such as exit amp criteria's, and even set up Amp limits for the Alternator.

A feature of the Smart Alternator Regulator when deployed in this way is the ability to protect the Alternator a bit more. It will limit the amount of Amps produced via the $SCV command. Both an Amp limit and the Watt limit may either be defined, or you can configure the regulator to auto determine the capabilities. Then limits can be made via the DIP Switch #7 (Use Small Alternator Mode) to configure the regulator to 'take it easy' on a smaller alternator. How easy can be changed by the derating values in the $SCV command.

There are a couple of new challenges with this deployment when trying to decide if it is time to exit Acceptance Phase:

- Other charging sources supplying current to the battery
- Account of house loads that are being served by the Alternator in addition to recharging the battery.

in simpler deployments, such as when this is the only charging source, and perhaps when any house load is limited or relatively consistent and small, accommodations can be made by increasing the exit Amp criteria for Acceptance Phase. In more advanced situations, an external coordinator can inform the Alternator Regulator directly what the Batteries Amps are via the $EOR command, thereby restoring complete battery care.

Example 3: Systems installation of regulator *(Simplified Wiring)*

When connected to other device in a 'system' deployment one of the benefits is simplified wiring. Rather than routing individual sensing wires to the battery for voltage, current, and temperature sensing, that information may be delivered over the CAN using a technique of remote-instrumentation.

Remote-instrumentation is a very reliable and long term used method for reducing the wiring needs in many industrial applications. By having a device located at the battery that is sensing the voltage/current/temperature of the battery and then sending that information via the CAN to the alternator regulator, you wiring burden is reduced to one CAN cable as opposed to several discreet wires. And if you installation has more than one charging source (say, twin engines, or an alternator and solar) this reduced wiring benefit becomes even greater.

To take advantage of remote-instrumentation you will first need a OSEnergy compliant device at the battery which senses battery voltage/current/temperature. Then when installing the Alternator Regulator you only need to connect sensing wires locally to the alternator saving long wires back to the battery.

At minimum you need to connect the VBat + and Vbat – wires to the local alternator + and – output, as shown in Figure 8:

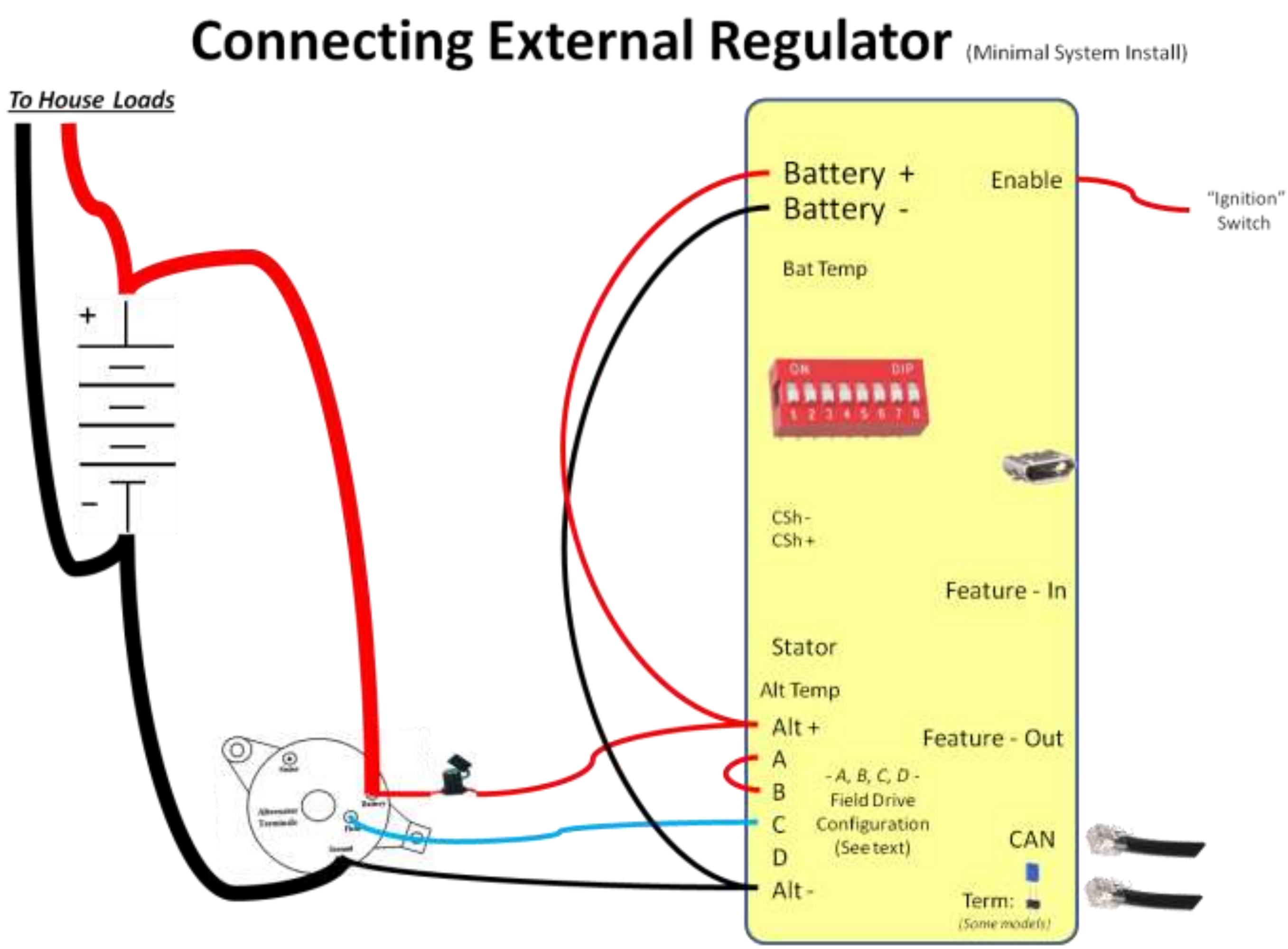

Figure 8 Wiring Diagram (Simplified Systems install)

EXAMPLE 4: SYSTEMS INSTALLATION OF REGULATOR *(ALTERNATOR INSTRUMENTATION)*

By taking advantage of remote instrumentation as shown in the example above, we can then use the remaining Alternator Regulator ports to monitor the alternator its self, using the wiring as shown in Figure 9:

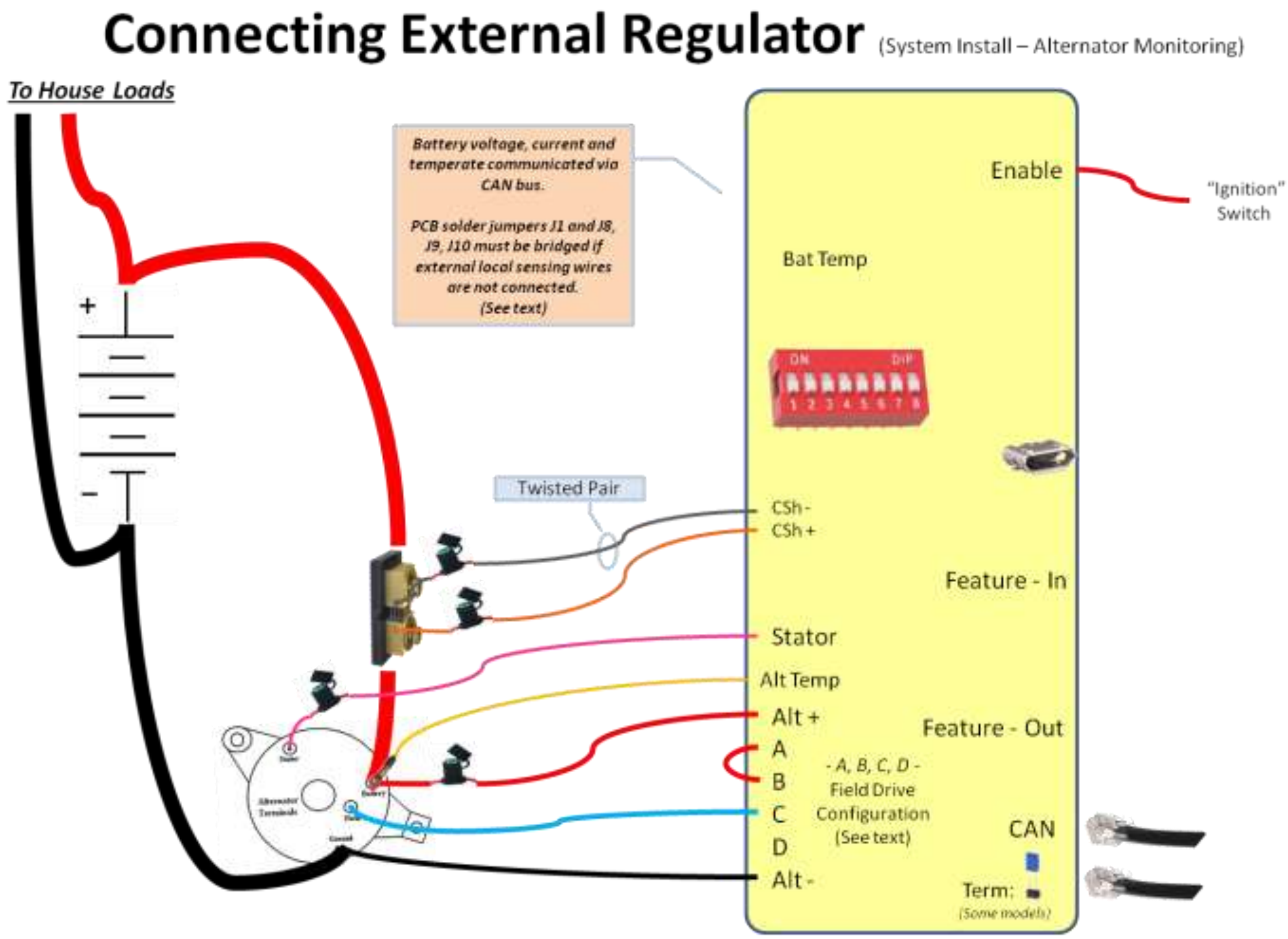

Figure 9: Wiring Diagram (Systems install with Alternator Monitoring)

This will give you the most benefit from your Alternator Regulator as it most closely follows a fundamental concept of OSEnergy 'Systems': Manage and control your local device – so in this case we are managing the alternator. You will notice that there are no voltage sensing wires. An option for the Alternator Regulator is to make the connection from VBat+ and VBat- internally, as opposed to needing external wires as shown in Figure 8. by 'Solder Bridging' PCB jumpers J1 & J8..J10 . Doing so will pick up VBat+ from the CH+ connection, and VBat- from the Alt- connection. These two photos show the jumpers soldered bridged J1 and J8, J9, and J10 respectively.

If you do not make use of the Solder Bridge jumpers, you need to manual connect the voltage sensing wires as shown here:

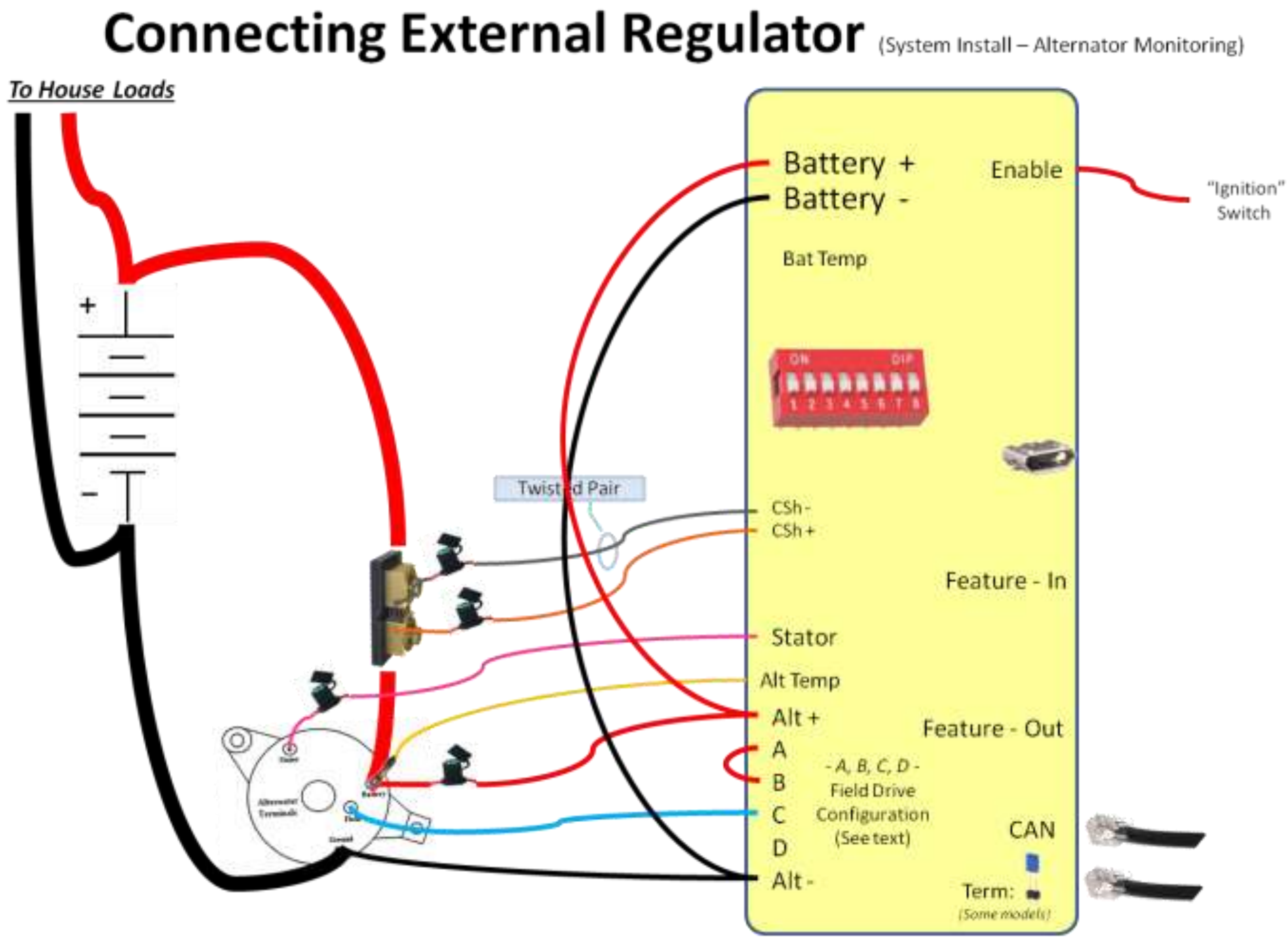

Figure 10 Alternative Wiring Diagram (Systems install with Alternator Monitoring)

EXAMPLE 5: SYSTEMS INSTALLATION OF REGULATOR *(HIGH RELIABILITY)*

There are a several options one may chose to do when installing the Alternator Regulator in a connected System to either increase reliability or a simplify installation to its basic. In Figure 11 below the voltage sensing wires have been routed and connected to the battery, as well as the battery temperature sensor. Further, the current shunt has been located to measure ALL current into and out of the battery (battery centric). With this connection the regulator is able to measure the status of the battery its self in the case of a failure in the remote battery BMS. It can also act as the remote instrumentation source for other OSEnergy compliant devices. The downside is the loss of the ability to report out alternator current, as the regulator is only able to see the total battery current.

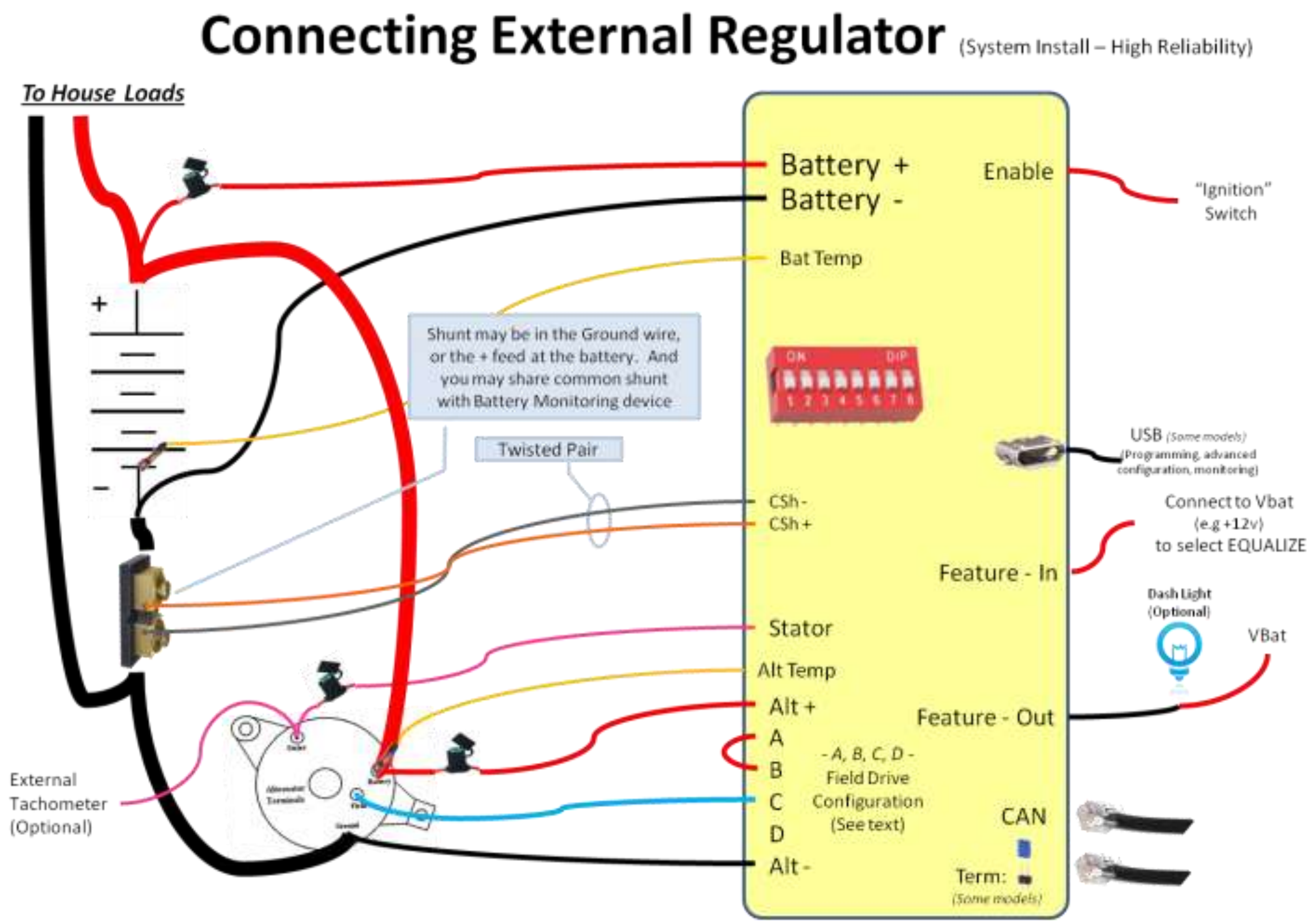

Figure 11: Wiring Diagram (High Reliability Systems install)

If you install the Alternator Regulator in as a High Reliability device in an OSEnergy system as shown in Figure 11, make sure to also configure the Alternator Regulator to indicate that it 1) Is allows to participate as a RBM, and 2) that the current shunt is indeed located at the battery (as opposed to on the alternator). Reference the ASCII command *$CCN* on page: 74 for details.

WHY DO WE NEED A STATOR WIRE?

The Smart Alternator Regulator includes a wire to be connected to one of the Status terminals on your alternator. Though optional, attaching this wire will improve the accuracy of voltage measurements, and enable several protection features.

- Improved Voltage Measurement by synchronizing the sampling of battery voltage with the stator. This is a 'best Practice' for measuring battery voltage as it allows us to see the true 'high point' of the alternators output each time we measure battery voltage. It also reduced external noise and other issues if the voltage samples was at random time. The Smart Alternator Regulator measured battery voltage to a resolution 1.25mV (a bit more then 1/1,000 of a volt), it is capable of very accurate sampling. By using the Stator wire we are able to maximize this accuracy.

- Connecting the Stator wire is also needed to facilitate Tach Mode. And if you then set the appropriate calibration values (pulley size, alternator poles, etc), you will be able to measure the engines RPMs via the ASCII status strings.

- Stator Sample allows for Adaptive Idle pullback, where the alternator load is reduced as the engine approaches idle. This can be helpful with small engines to prevent stalling, or sluggish performance near idle while still allowing for full current output at higher RPMs. (See $SCA: command 'PFB' – Pull Back Factor)

- Stator Sample also enables a few protection features. Example, if at some time we see SOME stator pulses, and then the disappear, we assume the engine has stopped. In this contrition we reduce the Field PWM drive greatly; without the stator wire we have no idea this has happened (remember, ANOTHER engine might be charging the same battery, so looking for a drop ion VBat is no use) and will just sit there applying field current and heating up the alternator.

Special note: If your existing installation has additional lamps, resistors, diodes etc. connected to the Stator field, or perhaps a Diode Trio (part of the dash lamp) or 'exciter' connection on the alternator, it is advisable to remove these. You can reconnect the dash lamp to the Feature-out connector. Leaving existing resistors, diodes, or other connections from the old installation have been known to cause issues with RPM measurements and other features of the Smart Alternator Regulator.

CONFIGURING THE SMART ALTERNATOR REGULATOR

The simplest way to configure your Smart Alternator Regulator is via the DIP switches. With these you can select one for the default Charge profile Entries, as well as tell the regulator how large of a battery you have (needed to more accurately decide when the battery is full).

USING THE DIP SWITCHES

Many of the basic features of the Smart Alternator Regulator can be configured using the DIP switched. Advanced changes may be made using the ASCII serial (See: *Appendix B: Sending data TO the regulator:*). The following table defines how the DIP switches are used:

Position	Meaning
1	On – Supply power to Bluetooth adapter Off – Disable Bluetooth
2..4 <4> <3> <2> Off, Off, Off Off, Off, On Off, On, Off Off, On, On On, Off, Off On, Off, On On, On, Off On, On, On	Select Charge profile 1..8 1 = Default (Safe) & AGM #1 2 = Flooded Lead Acid #1 (Starter type , etc) 3 = Flooded Lead Acid #2 (HD - Storage type) 4 = AGM #2 (Higher charge voltages) 5 = GEL 6 = Reserved for Future Use 7 = Custom #1 (Changeable – Preconfigured HD Storage with Overcharge) 8 = Custom #2 (Changeable – Preconfigured: LiFeP04) *(See* Table 3 *for more details)*
5,6 <6><5> Off, Off Off, On On, Off On, On	Define Battery Capacity as: 1x, – 500Ah 2x,, 500Ah – 1,000Ah 3x, 1,000Ah – 1,500Ah 4x. 1,500Ah and above
7	On – Use Small Alternator Mode Off – Use Large Alternator Mode Small Alternator Mode will restrict the maximum alternator output to 75% of its amperage capability. Large Alt mode limits output to 100%. See $SCA: command to change these values.
8	On – Tach Mode enabled Off – Disable Tach mode Some Diesel engines use the alternator to drive the tachometer. With these if the Alternator stops charging (say during Float mode), the tachometer can become sporadic or even stop working. Enabling Tach mode will cause the regulator to ALWAYS provide some small level of field drive, sufficient to provide a signal to the tachometer, but low enough to minimize Battery overcharge.

Table 1: DIP switch (2^{nd} Generation Regulator)

Position	Meaning
1..2	Battery ID The ‘Battery ID’ this regulator is attached to.
3..5 <5> <4> <3> Off, Off, Off Off, Off, On Off, On, Off Off, On, On On, Off, Off On, Off, On On, On, Off On, On, On	Select Charge profile 1..8 1 = Default (Safe) & AGM #1 2 = Flooded Lead Acid #1 (Starter type , etc) 3 = Flooded Lead Acid #2 (HD - Storage type) 4 = AGM #2 (Higher charge voltages) 5 = GEL 6 = Reserved for Future Use 7 = Custom #1 (Changeable – Preconfigured HD Storage with Overcharge) 8 = Custom #2 (Changeable – Preconfigured: LiFeP04) *(See* Table 3 *for more details)*
6,7 <7> <6> Off, Off Off, On On, Off On, On	Define Battery Capacity as: 1x, – 500Ah 2x,, 500Ah – 1,000Ah 3x, 1,000Ah – 1,500Ah 4x. 1,500Ah and above
8	On – Use Small Alternator Mode Off – Use Large Alternator Mode Small Alternator Mode will restrict the maximum alternator output to 75% of its amperage capability. Large Alt mode limits output to 100%. See $SCA: command to change these values.

Table 2: DIP switch (3rd Generation Regulator)

Using ASCII commands

In addition to the DIP switches, many parameters of the Smart Alternator Regulator can be customized by sending in a series of ASCII commands either via a Bluetooth connection, or by using an attached USB ← → TTL adapter board attached to the Service Port.

Changes sent to the Smart Alternator Regulator are saved in FLASH memory on the CPU for use next time the regulator starts up. (See Sequence of Operation section below for more details). Because of this you will need to restart the regulator after you have finished sending it ASCII change commands. There are some exceptions, ala the $FRM Force Regulator Mode command, $EBA – External Battery Amps, and a few more. See details of each command below.

You can restart the Smart Alternator Regulator either by cycling the power, or sending a $RBT command to the regulator.

Special note for 2nd generation Alternator Regulators and command lockout.

With the ability to support an RF Bluetooth module there is a security risk that someone could remotely alter the configuration of your Alternator Regulator. In order to increase the level of security, ASCII commands are disabled by default until you change the regulators name and password/PIN at least once via the $SCN: command.

This is to protect you from someone inadvertently, or maliciously, altering the configuration of your regulator. ASCII commands are also ignored if the system has been 'locked out' (See $SCO command)

See additional details in section 'Regulator Name & Password: need to Initialize' on page# 31

Altering Source Code

As a final way the source code may be directly modified, recompiled and then flashed into the Smart Alternator Regulator. This gives the ultimate flexibility, as well as the ultimate responsibility.

You will need the Arduino IDE downloadable from http://Arduino.cc The source code as well as support libraries used may be found under the Files link on the Smart Alternator Regulator blog. Make sure to also download included libs and files from the Files lnk, ala the more robust I2C library. http://arduinoalternatorregulator.blogspot.com/

Depending on the regulator version you have you will need to include the board type in the Arduino Board Manager. These are also in the Files link on the blog. You will also need to configure your Arduino IDE to recognize the 3.3v / ATmega328 opti-boot loader or the ATmega64M1 CPU used on the 3rd generation regulator. For more details, refer to the blog: http://arduinoalternatorregulator.blogspot.com/2010/06/assembly-and-programming.html

Connect your computer to the regulator, see: 'Communicating with the Smart Alternator Regulator' on page# 29

CHARGE PROFILES

Profile #	Battery Type	Bulk / Absorption Target Voltage	Exit Absorption when either:		Overcharge (Finish Charge)			Float Voltage	Equalize		Temperature Compensation (mV / 1f from 77f)
			Amps drop to	or Time exceeds	Target Amps	Exit Voltage	Max Time		Target Voltage	Max Time	
#1	**Safe / AGM-1**	14.1v	5A	6 Hrs				13.4v			13.2mV
#2	**FLA 1** (Start)	14.8v	5A	3 Hrs				13.5v			16.8mV
#3	**FLA 2** (GC, L16+)	14.6v	5A	4.5 Hrs				13.4v	15.3v	3 Hrs	16.8mv
#4	**AGM-2**	14.6v	3A	4.5 Hrs				13.4v			13.2mv
#5	**Gel**	14.1v	5A	6 Hrs				13.5v			16.8mv
#6	Reserved for Future Use										
#7**	**FLA 3** (GC, L16+)	14.4v	15A		15A	15.3v	3 Hrs	13.1v	15.3v	3 Hrs	16.8mv
#8**	**LiFePO4**	14.4v	3A	0.5 Hrs				14.0v			0mv

Table 3: Default Charge Profiles

All values are normalized for 12v / 500Ah battery and assume the Amp shunt is installed at the battery. If the shunt is mounted at the alternator, adjust the Exit_amp values to account for house loads. (A suggestion is to add 5A to the values shown in the above table.) All Amperage exit values will automatically scale up by the Battery Capacity Dip switch and likely match larger batteries with larger 'house loads'.

Blank sections indicate that feature is disabled.

** Profile #7 and 8 may be modified via the Change Profile ASCII commands: *$CP_:n*

See Appendix D: Details of CPE (Charge Profile Entries) for more details.

CHARGING LIFEPO4 BATTERIES

As I write this section during the Summer of 2015, LiFeP04 batteries (and variations) are starting to see deployment in house-battery applications.   There is a lot to be said: lighter, smaller, greater acceptance current capability (for shorter recharging times), and much wider usable discharge range then lead-acid based technology – combined with lower costs bring this new technology into a very viable candidate.

However, it is also true that most deployments of LiFeP04 technologies have been in Electric Vehicle's – not house storage battery applications.   And as such, what is 'known' these days are shaped largely by experience in a vastly different usage model.

- Vehicle usage = High Discharge rates, followed by long recharge cycle with little to no parasite loading.
- House usage = Moderate to low discharge rates, over a longer period of time. Recharging often occur concurrently while moderate to low 'house loads' loads are still present.

Vehicle usage focuses on a simple charge profile: Recharge until Vpc reaches a known threshold and then turn off all charging sources.   However, such an approach is suboptimal for house usage. Specifically it precludes the ability for a charging source to carry house loads after the battery had been fully recharged. Not allowing for this mode (Float if you will) can result in arriving at your designation, or entering the night, with a less then a fully charged battery.

There is still a lot to learn (and with that, unlearn) as experiences with LiFeP04 technologies advances.

And this presents us with some options when deploying the Smart Alternator Regulator in conjunction with LeFeP04 batteries.   Options imagined today may change as more is learned, fortunately the great flexibility contained allow for those changes to be easily deployed. The following are some concepts to consider when integrating the Smart Alternator Regulator into a LiFeP04 (or like chemistry) based system;

One of the 1st decisions is where the control point will be. Most LeFeP04 batteries are deployed in conjunction with a BMS, and many of these BMS devices have a signal that is indented to enable or disable charging sources. Other capabilities often include safety warnings and/or disconnects to prevent over chagrining, or over discharging; health of the battery, and even active cell balancing. There are perhaps two ways to enable communications between an external BMS and the Smart Alternator Regulator – both use the Feature-in port and CPE #8 where Feature-in is used to force the regulator into Float mode when active, and force it back into charging mode (Bulk) upon going inactive.

Consideration 1: BMS is control point

- BMS makes all decisions as to if the alternator should be charging or not.
- Regulator CPE #8 is adjusted to set its voltage points slightly outside the envelope the BMS uses to make charge / halt-charging decisions – but at levels less than the BMS disconnect voltage levels. Example, if a BMS has the following two voltage set points:
  - Charging Completed: 13.8v
  - Fault charging disconnect voltage 14.4v

  One might set the CPE's Bulk/Acceptance voltage level to 14.0v. In this way during normal operation the BMS can signal the alternator to go into Float mode at 13.8v, but if something happens to prevent that the regulator will its self transition at 14.0v before the BMS faults and disconnects charging sources.

- Likewise, voltage levels for Float_to_bulk can be set slightly lower than the BMS 'resume charging' level, but again slightly above any BMS disconnect for excessive discharge.

Consideration 2: Smart Alternator Regulator is control point:

- Much like the above, however in the case the basic roles are swapped, the regulator decides when to begin charging while the BMS is used as a backup.

Consideration 3: What to do about Float

- Much of the knowledge in existence today has been derived from EV (Electric Vehicle) usage profiles.
  - In EV usage a discharge cycle is followed by a complete recharge cycle – and once fully charged the battery's charging sources are turned off.
  - Until the EV is used again, very little current is drawn from the batteries – so this approach works well.
- However, in house battery usage there is often a noticeable 'static' load to power instruments, refrigeration, and communications gear. Perhaps pumps, computers, any number of items.
- In this usage model it would be desirable to allow any charging source to support those house loads while that charging source is available – preserving the charge in a full battery.
- This brings up the question of what to do about Float, some ideas:
  - i. Disable Float by setting the float target voltage very low, perhaps just above the BMS excessive discharge voltage trip point – again as a backup.
  - ii. Set float to a natural voltage level for idle cells, example 13.4v for a 4-cell ('12v') battery
  - iii. Utilize the Amps regulation capability (this required the Amp Shunt to be installed on the battery) and set the Float CPE to 'regulate' battery current to 0A – neither allowing discharging of the battery, nor allowing additional energy to enter it.
  - iv. Perhaps a combination of ii & iii
- The default CPE #8 uses approach ii & iii, setting VBat to battery 13.36v max and regulating current to 0A.

As more is learned about these deployments some of the above may be refined, and/or augmented. You can always check the Blog to for additional insight.

http://arduinoalternatorregulator.blogspot.com/

FEATURE-IN

Feature In is enabled by connecting the Feature In pin a voltage greater than 6v. Voltages as high as 72v are allowed. Feature in currently serves three purposes:

1) Master reset: Holding this line to VBat before the power is applied, and then applying power will reset the controller to as-compiled (default) conditions.

2) With the Alternator Running, holding this line to VBat will enable EQUALIZE mode providing:
   a. The Regulator is currently in Accept or Float mode
   b. The current Amps being delivered are equal or less then the CAP Amps (if defined) for Equalize mode in the current Charge Profile. (Equalize should only be entered on fully charged batteries)

   Once EQUALIZE mode is entered, the regulator will remain in the mode until one of three conditions are met:

   1. FEATURE-IN line is no longer connected to VBat
   2. Defined exit Amps have been meet
   3. Defined timeout has been meet.

A comment on entering EQUALIZE mode: Equalize is considered a high-monitoring mode and there is a potential for damage to the batteries. As such, the Feature-in port must remain connected to VBat during this time, and the LAMP light will blink-out the equalize status pattern.

Option #1 and #2 are only active of CPE #1..7 is selected. If CPE #8 is selected, then only the following option will be active.

3) If CPE #8 is selected (default LiFePO4), holding this line to VBat will prevent the regulator from entering one of the ‘charging’ modes (Bulk, Acceptance, Over-Charge). Instead only Float will be allowed. This capability is of advantage when used in conjunction with an external Battery management System (BMS), common on LiFePo4 batteries, that feature a High Voltage Cutout, or Charge Enable signal.

   It should be noted that as long as Feature-in is held active, the Smart Alternator Regulator will remain in Float mode – ignoring any of the normal exit criteria (e.g., timed exit, or accumulated Ah exit). And that when transitioning from Active to Inactive, the regulator will immediately enter Ramp/Bulk mode. See the section ‘Charging LiFePO4 batteries’ above for more details on using this option.

Feature Out-

The Feature Out port is an open-collector driver and can be connected to a dash 'Alt' light - operating akin to a traditional ALT indicator lights.

Options for Feature-out: In the source code there are several optional uses of the Feature-out connector. These are enabled via selecting different #define statements, recompiling the source, and reflashing the firmware. See the source code for details, but currently the following options are included in the source:

- #define FEATURE_OUT_LAMP - Enable LAMP / Fault driver – DEFAULT OPTION
- #define FEATURE_OUT_ENGINE_STOP - Active when we enter FLOAT mode.
- #define FEATURE_OUT_COMBINER - Allows sharing of charging source(s) between two batteries.

Feature Out is active Low and is limited to 0.5A on the Feature Out connector. This is not only due to the thermal limits of Q3, but also that the return line (battery -) is shared by the Feature-out and the VBat voltage sensing. Placing too much return current via the Feature out connector can impact the accuracy of measured battery voltages.

Feature Out Lamp (default option)

In this mode the Feature Out port is useful for connecting to a dash mounted Alt lamp. The port will be disabled (lamp out), unless in one of the following conditions:

- Indication of no-charging status – lamp will be full on during this time. (e.g., during engine warm-up period)
- Fault status – lamp will blink-out the fault code (See status LED section)
- Controller resetting – Lamp will blink-out resetting LED pattern (See status LED section)
- Regulator is in EQUALIZE mode – Lamp will blink-out EQUALIZE LED pattern (See status LED section)

Feature Out Engine Stop

With this option selected, Feature Out will become active once the alternator enters Float or Post-Float mode. This can be used to signal an external generator that the battery is fully charged, and it is OK to stop.

Feature Out Battery Combiner Option

(The following description is for Source code v0.1.3 and beyond)

Many situations have more than one battery bank, but only one Alternator. Or they might have two batteries, each with their own alternator. An example might be a boat that has a large battery bank for the House battery, and a smaller battery used for the Starter, or a Bow Thruster. . Using the Feature Out to drive a high power relay is an inexpensive way to get a 'smart' battery combiner.

There are perhaps two common reasons for using a battery combiner:

1. Allow a 2nd charging source to 'help' recharge the large house battery during its Bulk phase
2. Recharging a 2nd battery which has no charging source of its own.

Because the Smart Alternator Regulator is designed to focus on the attached battery, and not the 'combined in', some assumptions and care must be made. We want to make sure we do not overload a 2nd alternator, or overcharge a 2nd battery. Some additional considerations which must be taken into account:

- Do not exceed the current limit of the Feature Out port (0.5a) with the coil draw of the external large relay.
- Use a length of medium size wire (e.g., 10' of 10g wire) to create a level of resistance between the two batteries. A common min resistance of 5-10mΩ appears to be common (Milli-Ohm)
- Monitor smaller alternators not managed by the Smart Alternator Regulator - watching for signs of overheating.

Situation #1: Using 2nd battery/alternator to 'help' the house battery during Bulk.

In this example there are two fully independent battery and associated charging systems. And example might be the factory alternator and starter battery, and a house battery with its own alternator using the Smart Alternator Regulator. The starter battery will likely be quickly recharged after the engine starts, leaving a significant amount of unused capacity in the starter battery's alternator.

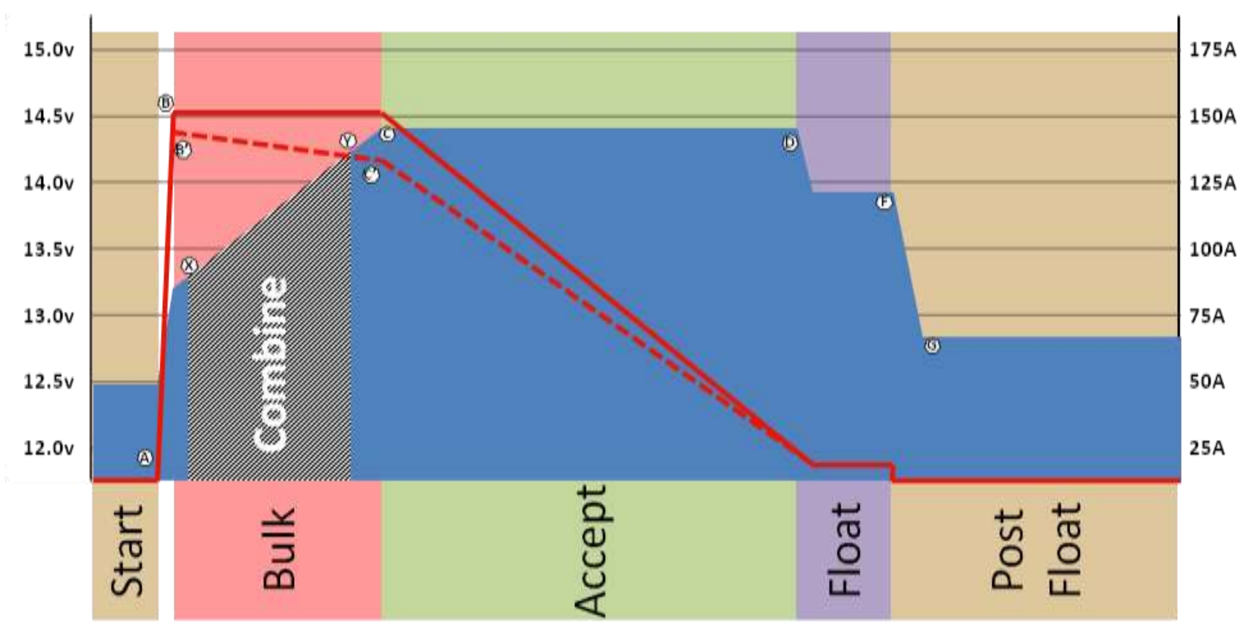

Figure 12: Receiving Help combiner profile (default)

In this mode, Figure 12 above illustrates when we want to combine the two systems (between points 'X' and 'Y'). Specifically during Bulk we want to:

- Wait until the house battery reaches 13.2v before enabling the combiner. If we combined sooner there is a risk of pulling energy from the 2nd battery, as opposed to only asking the 2nd alternator to share its capacity. 13.2v will also reduce the voltage difference between the two batteries thereby minimizing initial surge current.
- Break the connection of the two batteries after 14.2v. With the assumption the 2nd battery has its own charging source that will handle all the recharging needs of that battery, we do not want to 'override' those decision. 14.2v was selected under the assumption that many 'starter' batteries are connected to a default internally regulator fixed voltage alternator; those are often in the 13.8 to 14.2v range.

<u>Situation #2: Recharging a 2nd battery which has no charging source of its own.</u>

In this example there are still two independent battery, but only one charging source. A representative example would be a house battery / alternator being controlled by the Smart Alternator Regulator, and a 2nd battery used perhaps to power a bow-thruster, or even a starter where there is no 2nd alternator. Unlike the case above were we are looking to gain assistance from the other battery/alternator, in this case we are asked to be the charging source for the 2nd battery.

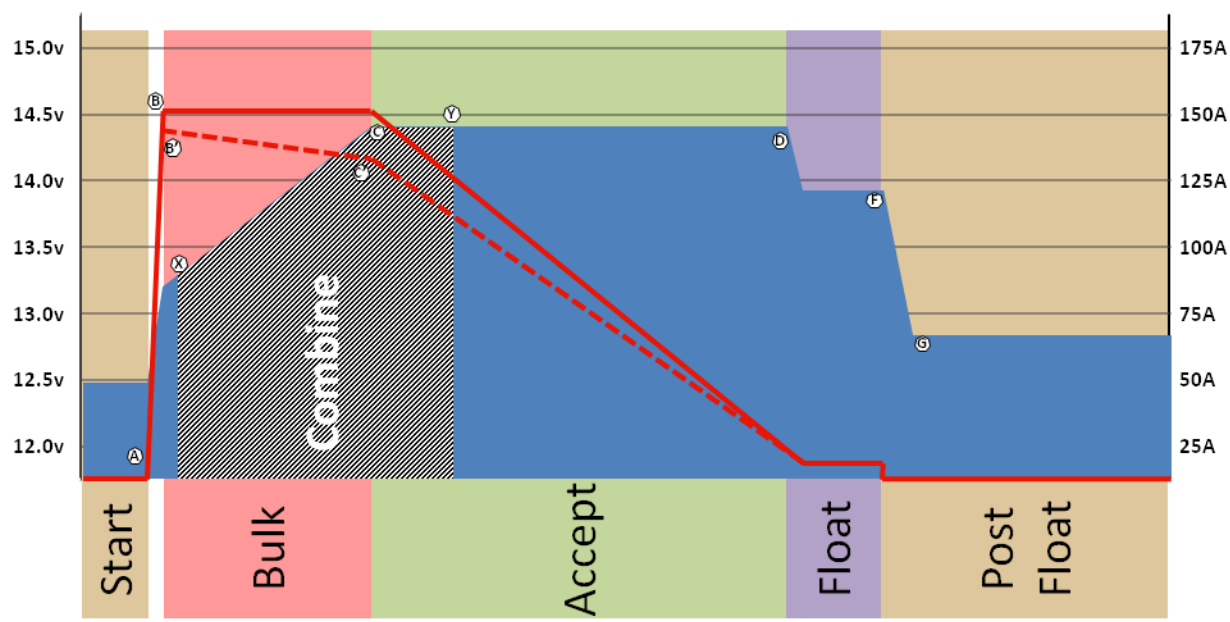

Figure 13: Charging 2nd battery combiner profile

Figure 13 illustrates the conditions when we want the combiner enabled. The idea is the 2nd battery parallels the charge profile of the main battery, and because we are not actively managing the 2nd battery we cut off the Acceptance phase after a relatively short time (Points 'C' to 'Y' above) to reduce the risk of overcharging the 2nd battery, in effect boiling it off.

Care must be taken in each of these situations to protect the 2nd battery and charging system; remember the Smart Alternator Regulator will focus on its battery and adjust things to its needs with NO regard to the other batteries needs - outside of the limited combiner configuration options.

Combiner Configuration Options

All options for configuring the Feature-out port as a combiner are made in the source code, there are no ASCII commands to modify these.   The following variables in the source code:

```
#define     COMBINE_CUTIN_VOLTS            13.2
#define     COMBINE_HOLD_VOLTS             13.0
#define     COMBINE_DROPOUT_VOLTS          14.2
#define     COMBINE_ACCEPT_CARRYOVER       0.75*3600000UL  // ¾ an hour, 45 minutes.
```

COMBINE_CUTIN_VOLTS when the combiner is enabled (point 'X' below), along with COMBINE_HOLD_VOLTS. Once voltage raises to the CUTIN level (13.2 by default) the combiner will be enabled (via Feature-out), and will stay so even if the battery voltage temporarily dips – as long as it remains above the HOLD level (13.0v by default).. However, if it drops too much and goes below the HOLD level, the combiner will be disabled.

As battery voltage rises, the combiner will be disabled by either rising above the DROPOUT level – point 'Y' shown in Figure 12 (14.2v by default), or after being in the accept phase for CARRYOVER duration – point 'Y' shown in Figure 13. Either way (along with low voltage) will cause the combiner to open.

Note that as with all Charge profile Entries, voltages shown above are 'normalized' to a 12v battery and will automatically scale by the system voltage multiplier. (e.g., in a 48v system, the multiplier is 4, so the combine voltage becomes 4 * 13.2v, or 52.8v).

,

COMMUNICATING WITH THE SMART ALTERNATOR REGULATOR

The Smart Alternator Regulator supports external communication of status and more advanced configuration changes then the DIP switched allow for. Communications is primarily via Serial ASCII strings, the formats of which are documented in *Appendix A: Receiving data FROM the regulator:* and *Appendix B: Sending data TO the regulator:*   The 3rd generation regulator adds support for Control Area Network (CAN) communications. Refer to *Appendix C: CAN messages* for additional details of the CAN communications protocol.

A special note, some terminal programs do not send a complete end-of-line terminator (Notable, some Arduino IDEs). To support these environments, the Alternator Regulator will recognize the character '@' as an alternative EOL.

2nd Generation communications: Physical connection is made either via the SERVICE port located on the regulator, or wirelessly via the included Bluetooth module using SPP (Serial Port Protocol).   Remember changes will be blocked until the NAME and PIN code have been updated; see 'Need to Initialize' in the Bluetooth section on page 31 and the $SCN: command in Appendix B.

The Service port is a 6-pin female header located in line with a 2-pin expansion connector. Pin 1 of the service port is on the end of the female socket located next to the two male expansion pins. The Service port is defined as:

	Pin #	Purpose
■	6	+3.3v
■	5	GROUND
■	4	n/c
■	3	RX: Regulator Serial Receiving pin (Overrides Bluetooth module)
■	2	TX: Regulator Serial Transmission pin
■	1	~RESET: Resets regulator on falling edge
●	SLC	I2C Expansion (male pin - 3.3v)
●	SDA	I2C Expansion (male pin - 3.3v)

Figure 14: Service Port pin-out (Generation 2 regulator)

Serial communication is at 9600 Baud, 8-bit, no parity, 1 stop bit. **When the Serial port is attached it will have hardware priority – preventing any communications *<u>received</u>* via the Bluetooth from reaching the regulator.** You will still be able to monitor the Smart Alternator Regulator status via Bluetooth – just not send any commands via Bluetooth. The 3.3v pin can be used to supply limited power to an external expansion device, or optionally to power the Smart Alternator Regulator when it is not installed.

Remember that the 2nd generation regulator will lockout all ASCII commands until the NAME and PASSWORD has been updated from its default. See section 'Regulator Name & Password: need to Initialize' on page# 31

Service Port pin outs are in an order to match a very select USB ←→ TTL adapter, example photos here:

Take careful note of the pinouts, there are a wide verity of 6-pin ones. I found mine on EBay using the search: "CP2102 USB 2.0 to TTL UART Module 6Pin Serial Converter STC Replace FT232" When verifying the pinouts, look closely at the 3.3v supply and the DTR signal in position 1 (used to drive ~RESET)

3rd Generation Regulator: The 3rd generation regulator features a built in USB connector. Simple connect a micro USB cable to the connector and a serial terminal program of your choice. The default communications rate is 115,200 Baud, 8-bit, no parity, 1 stop bit.

Serial commands may also be carried over the CAN bus using the J1939 CAN 'Terminal' communications protocol DGN (17E00h)

Bluetooth Operation

Optional to the 2nd generation regulator, serial communications may be initiated with the Smart Alternator Regulator using Bluetooth. Attach an external computer, tablet, or smart-phone using the SPP (Serial Port Protocol). Then use a serial terminal to see the ASCII strings, or send commands to the regulator. The default NAME and PASSWORD for the Bluetooth connection is:

NAME: "ALTREG"

PASSWORD: "1234"

Regulator Name & Password: need to Initialize

Each Alternator Regulator has a name associated with it. This name is used to help identify the regulator when communicating and is used for the Bluetooth ID, as well as the CAN identifier. Each regulator also contains a password which is primary used by the optional Bluetooth module. Both the NAME and PASSWORD may be any combination of alpha numeric character up to 18 characters long - with the exception of the comma (,) space ( ) or the '@' character.

After a master reset (or when 1st using the regulator) you will be able to view the Smart Alternator Regulator status via ASCII strings. However because the ability to change the configuration could potential harm your battery **the ability to SAVE any configuration changes is disabled until the Name & Password has been updated via the $SCN: command.**

For additional security - to *initially* change the Alternator's Name / Password you MUST set the DIP Switch 1 = OFF (Bluetooth disabled), and DIP switch 2..8 = ON and the alternator must not be charging (e.g., engine is powered on but stopped) . Once the name/password has initially been updated you can then restore the DIP switches to the configuration needed for your system; and the name/password may be changed again without having to change the DIP switches to all on. (unless you reset the regulator to defaults 'as compiled' condition – see *Restore to AS-Compiled (default) Status* on page 39)

Caution: The regulator will NOT check the validity of the strings, specifically that they do NOT contain a comma (,) space ( ) or '@' character. Be careful not to send these as the results might be unpredictable. If needed, perform a master reset to the regulator and start over.

It is STRONGLY suggested you change BOTH the NAME and PASSWORD to something unique OR disable the Bluetooth via DIP-switch #1. And remember: <u>no ASCII changes will be accepted by the Smart Alternator Regulator until you do</u>. Else you run the risk of someone near you accidentally, or maliciously, adjusting your Alternators configuration and potential damaging your system / batteries. (And this is why the ability to save configuration changes is disabled until the 1st $SCN: command has been received, and that can only happen if you set all the DIP switches to the ON position.)

Operation Overview

The Smart Alternator Regulator has a wide range of flexibility and can be configured in three ways, ranging from Simple to Any-thing goes.

- Select among built in (default) parameters via the DIP switches
- Modify many of the built in parameters via sending ASCII commands over the Service port and/or the Bluetooth serial connection
- Modifying any or all built in parameters in the source code, recompile and re-flash to regulators firmware. (requires use of Service port and Arduino IDE, see http://arduino.cc/)

Startup Sequence:

When the Smart Alternator Regulator is first powered up it loads in the default Firmware and then looks to the DIP switches to see what the user has selected. It will then sample the system voltage to decide if it is installed in a 12v, 24v, 32v, or 48v environment (and thereby setting the SysVolt multiplier, see SST: / System Status below for more details about this and how it is used). Finally it will look at the saved FLASH memory in the CPU to see if any of the parameters have been modified or any of the DIP switches have been overridden by the user using ASCII commands.

Note that parameters saved to Flash are ONLY checked during startup; this is why after changing any parameters via an ASCII string command you need to reset the regulator (via the $RBT – ReBooT command) in order for any of those parameters to take effect. This is done this way so that the regulator does not start acting on changes part way through you making configuration changes, ala, if you are updating the CPE entry, you might want to see ALL parts of it before the regulator starts using the new values.

Ramp-up Sequence:

Next the Smart Alternator Regulator will enter the Ramp-up phase. It first waits 30 seconds before doing anything (see #define ENGINE_WARMUP_DURATION in the source code) to give the engine time to start. Next it will begin to slowly ramp up the alternator field over a period of about a minute before entering Bulk phase.

Charging the Battery:

Once the regulator has completed its startup and ramping sequence it enters the ‘charging’ phases. Each phase has limits (ala voltage, amps, etc) as well as ‘exit’ criteria based on voltage, amps and/or time. A high degree of configurability allows each phase to be used, or bypassed as needed to meet the battery manufactures recommended charging profile. While in each of these phases the regulator will also monitor battery and alternator temperature, making adjustments as required. It will also look to see if the regulator has been configured with System limits and make sure none of those are exceeded.

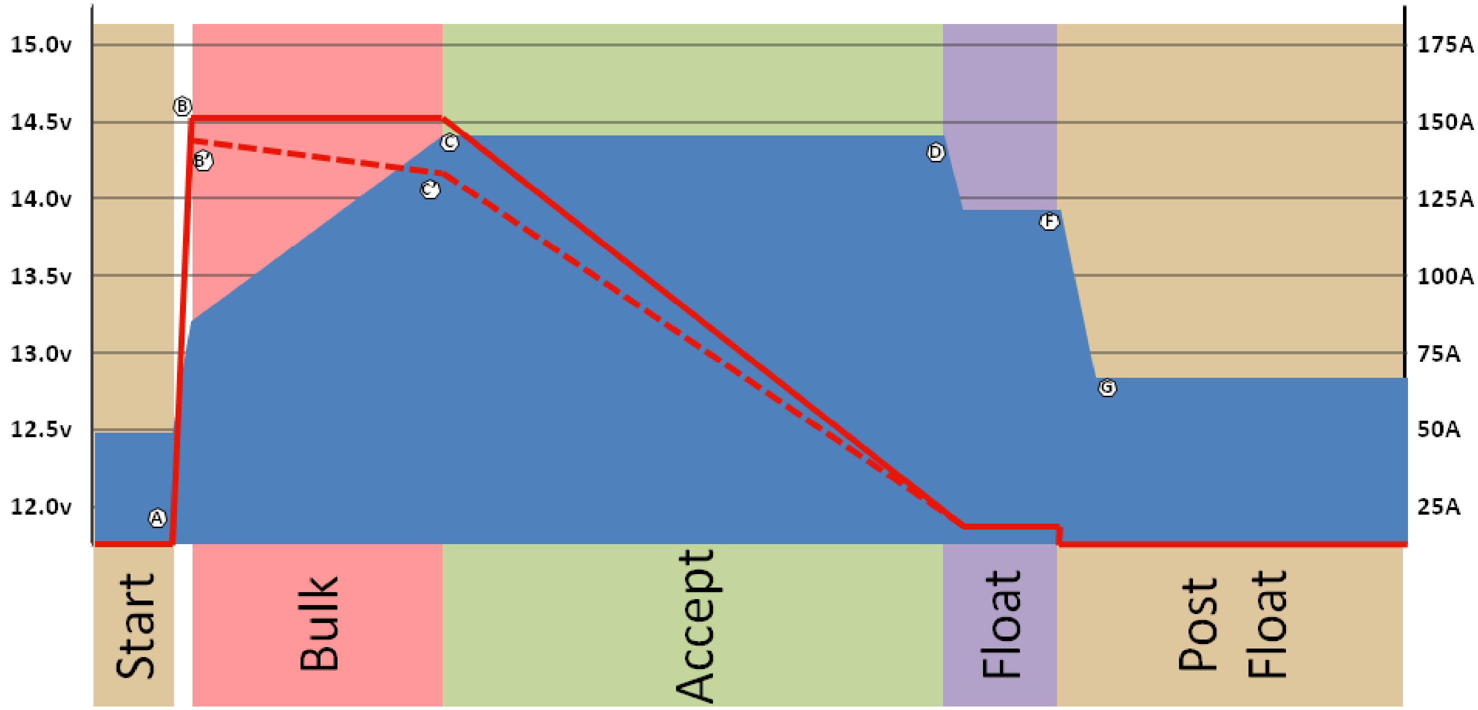

Figure 15: Example 3-stage Charge profile

Figure 15 above shows a typical 3-phase charge profile, consisting of the Bulk, Acceptance, and Float state. This is one of the simplest charge profiles, and very commonly seen in external regulators. The following table details each phase as well as the parameters in the source which can impact them.

Phase	Ref	Limiting Factor	Exit Criteria	Discussion	Key Source Variables
Start	-A	Time	60 second delay or Stator Signal.	Allows time for the engine to start and get up to speed before we apply a load to it. If Tach Mode is enabled, and the regulator has been configured with a fixed minimum PWM value via the *$SCT:* ASCII command (see: Tach Min Field), Start Phase will be extended past the 60 seconds initial delay until a value Stator signal is detected.	ENGINE_WARMUP_DURATION

Phase	Ref	Limiting Factor	Exit Criteria	Discussion	Key Source Variables
Ramp	A-B	Time	Exceeding xx seconds Reaching target Voltage	Soft ramp of applied load. Normally this will exit via the Time variable, but if the battery is already well charged it will very quickly reach the target voltage set point, forcing an exit of Ramp(A-B) phase	PWM_RAMP_RATE ACPT_BAT_V_SETPOINT
Bulk	B-C	Alternator Capacity	Battery Voltage Time(optional)	Using default configuration (Battery Focused). During this phase the majority of energy is returned to the battery, often touted as up to the 80% SOC point. The Alternator is run full out producing as much power is it is capable of. As the battery recharges, voltage will raise until the set point is reached which will trigger the exiting of Bulk. If Small-Alt mode is selected via the DIP switches, then the output of the Alternator will be managed to a lower level, this is to prevent overheating of small alternators unable to sustain continuous max amp output. Likewise, if Half-power mode is selected (via shorting the Alt Temp probes together), an more aggressive scale back will occur. This can be useful in installations where extra engine power is needed, ala to drive an additional pump, or perhaps allow for high speed / power mode (passing)	ACPT_BAT_V_SETPOINT ALT_AMP_DERATE_NORMAL ALT_AMP_DERATE_SMALL_MODE ALT_AMP_DERATE_HALF_POWER
Bulk’	B’-C’	System Capacity	Battery Voltage Time(optional)	An alternative configuration (Alternator Focused) to manage the system load at a lower level than the full capability of the Alternator. This is useful when powering a large alternator with a small engine, ala a DC generator. All items discussed in Bulk above are still applicable in this configuration, with the additional system limit added. Notice in Figure 15 above: as the battery voltage increases – amps delivered are reduced resulting in a constant wattage. Which translates to a constant load placed on the driving engine.	ALT_WATTS_LIMIT ALT_AMPS_LIMIT

Phase	Ref	Limiting Factor	Exit Criteria	Discussion	Key Source Variables
Accept	C-D	Battery Acceptance Rate	Battery Acceptance Current Time – Fixed Time - Adaptive	Acceptance carries recharging from 80% to 100% of the batteries capability. The preferred way to exit this phase is via measuring the current the battery is accepting; indicating its state of charge. Other ways to exiting include a fixed Max time, as well as a time ratio based on a multiplier of how long the battery was in the Bulk phase. Both of these can be useful for installations where the Amp shunt has not been installed, or it has failed.	ACPT_BAT_V_SETPOINT EXIT_ACPT_AMPS EXIT_ACPT_DURATION ADPT_ACPT_TIME_FACTOR
Float *Forced Float*	D-F	--	Pull back to Bulk Time(optional)	Float is a 'parking place' intended to maintain a fully charged battery. A slightly higher than rest voltage is maintained to overcome the batteries internal self-discharge. Further, the Alternator is maintained in a ready mode to supply current to any external load that might be requested. If the Alternator is unable to support this load and current is requested from the battery, we are pulled out of Float and back into a recharge cycle (back to Bulk). Another indication of this is if the battery volts start to drop. This can be helpful in situations where the Amp shunt is not installed, or has failed. Finally, the accumulated Amp Hours may be monitored and once a given amount has been removed from the battery – the regulator will revert to Bulk mode. Note: If CPE #8 is enabled, the FEATURE_IN pin may be used to force the regulator into Float mode. As such, none of the above exit criteria will be checked while FEATURE_IN is active.	FLOAT_BAT_V_SETPOINT EXIT_FLOAT_DURATION FLOAT_TO_BULK_AMPS FLOAT_TO_BULK_VOLTS FLOAT_TO_BULK_AHS <NONE – All exits bypassed>

Phase	Ref	Limiting Factor	Exit Criteria	Discussion	Key Source Variables
Post Float	F-G	--	Pull back to Bulk Time(optional)	Post Float is much like Float, except the alternator is turned off allowing the battery to sit on its own. Post Float will return to Float mode based on a timed exit value. Once this has been exceed, the regulator will revert to FLOAT mode. Another way to exit Post Float is through the appearance of a large load which has been placed on the system. This can either be determined by battery voltage, or by monitoring the accumulated Amp Hours removed from the battery – much like Float above. Note however that when exiting via these conditions these two methods will cause the regulator to move back into Bulk mode - just like Float above.	EXIT_PF_DURATION PF_TO_BULK_VOLTS PF_TO_BULK_AHS

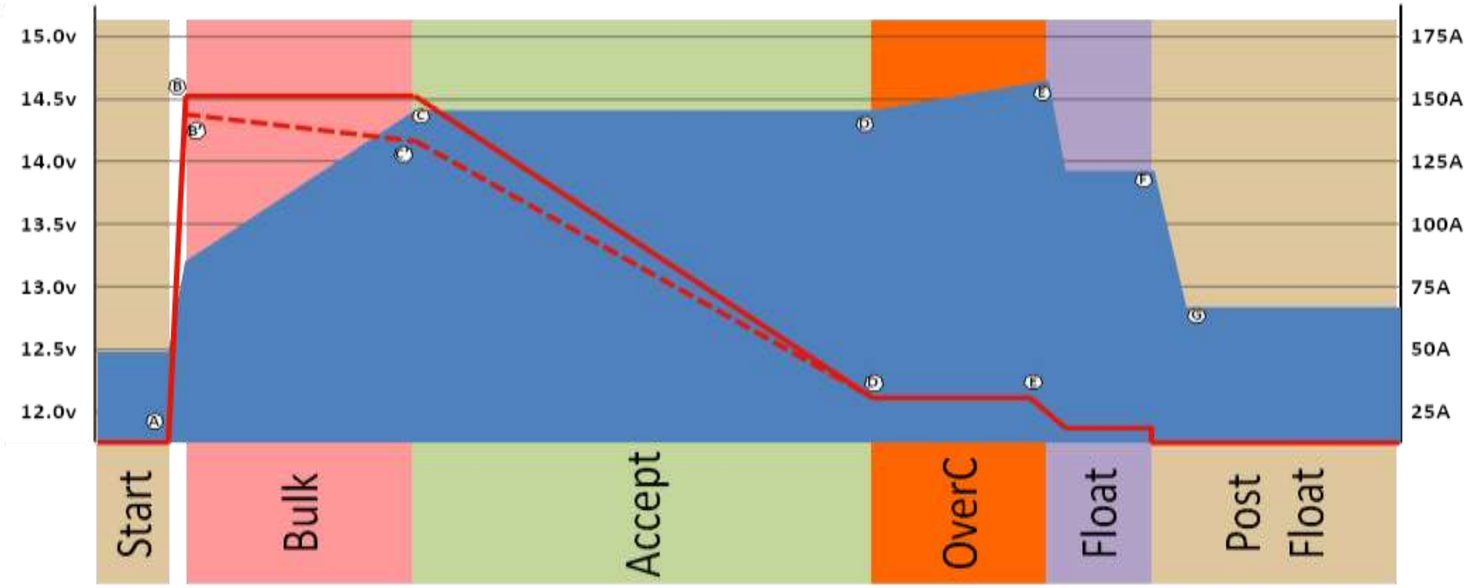

Figure 16: 4-stage Charge Profile

Some batteries ask for an additional charging phase after the Acceptance phase, one that provides a small constant current and allowed the voltage to raise. Referred to as a Finish Charge or an Over Charge, the purpose is to assure even chemical actions has occurred throughout the entire battery. For batteries which specify this 4th charge step, the above charge profile is used. Refer to the above table for stages other then Over Charge.

Phase	Ref	Limiting Factor	Exit Criteria	Discussion	Source Variables
Over Charge	D - E	Defined Amps	Battery Voltage Time(optional)	In this phase the alternator will be managed to maintain a consistent current into the battery, allowing battery voltage to raise (much like Bulk phase). Once battery voltage has reached the exit volts (Point E), or a max time has been exceeded, the regulator will move to the Float mode.	LIMIT_OC_AMPS EXIT_OC_VOLTS EXIT_OC_DURATION

Note how the Smart Alternator Regulator will move directly to Acceptance Phase if the battery voltage quickly raises. This, combined with the measurement of the Acceptance Current to determine if a battery is fully charged handles the all too common situation of you being plugged into shore-power and hence getting under way with a the battery already fully charged. Traditional time-only based regulators you might be stuck in the Acceptance phase for several hours until the fixed time limit is reached, overcharging the battery and boiling off water. With the Smart Alternator Regulator the regulator will recognize quickly the battery is fully charged and move into float mode within seconds of starting the engine. This eliminated perhaps one of the major ways a less intelligent regulator can damage your batteries.

Each of the charge steps are fully configurable, and many can be bypassed (ala, most charge profiles do not use the Overcharge step). As way of an example, the following two diagrams show two different approaches for charging a heavy duty 12v / 500Ah battery (ala, two L-16 batteries). Both profiles are included in the default Smart Alternator Regulator as profile #3 and #6. Figure 17 shows the traditional 3-step charge profile (Bulk, Accept, and Float), while Figure 18 illustrates the flexibility to accommodate a 4-step charge profile (Bulk, Accept, Over-charge, then float). The diagrams provide an overview of each phase and represent a 180A alternator charging a 12v 500Ah battery.

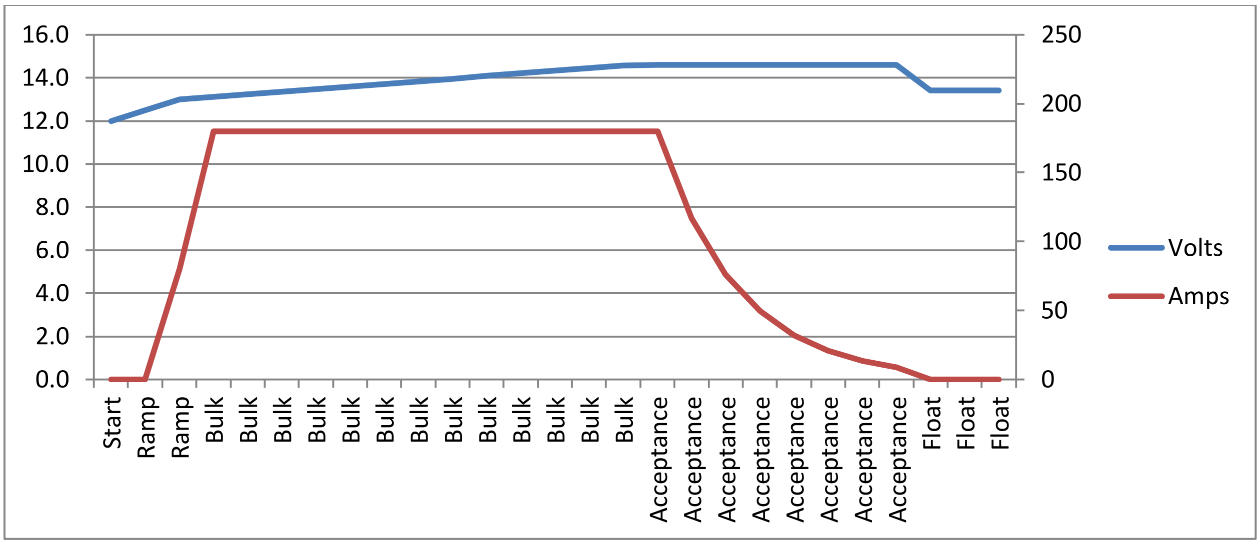

Figure 17 – Example Charge Profile #3 (HD FLA Storage battery)

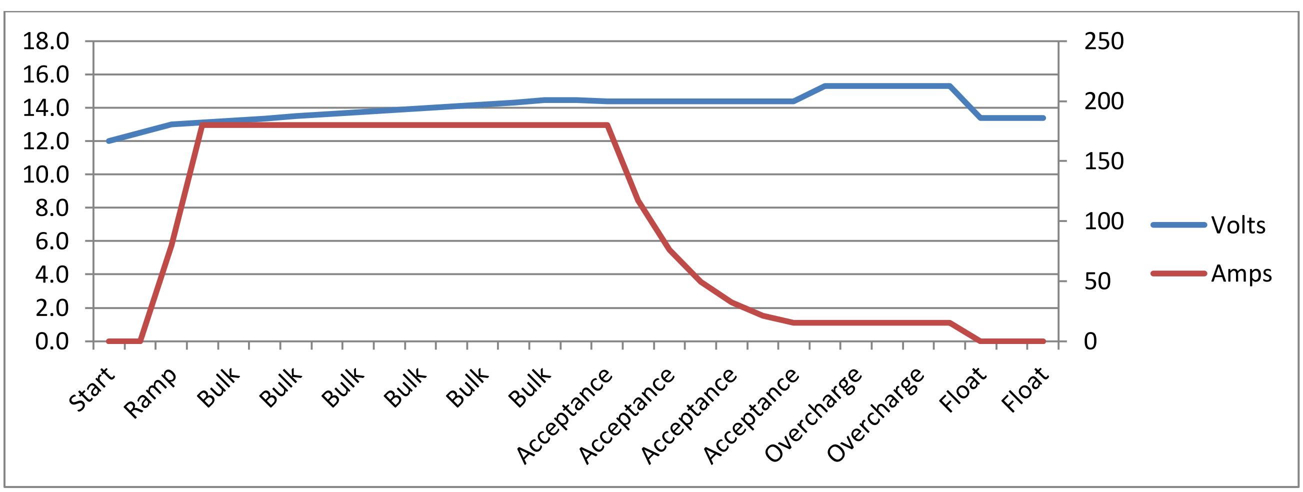

Figure 18 – Example Charge profile #6 (HD FLA - Alternative Charge Profile)

RESTORE TO AS-COMPILED (DEFAULT) STATUS

Configurations (system and Charge parameters) which have been changed from the as-compiled default values may be restore in one of three ways:

- Individually by the $CPR:n , $SCR: and $CCR: commands.
- Collectively via the $MSR: command.
- Total system restore by connecting the FEATURE-IN to VBat before applying power to the regulator, and then applying power. This will restore ALL configurations to the default condition.

Note that if the regulator has been Locked Out via the $SCO command, the only way to restore to the default configuration is via using the Feature-in capability, none of the '$' change commands will function once the regulator has been locked. You may also clear the locked flag in the FLASH by recompiling the firmware and reloading it via the service port. If you do this, you will need to make a change to the following defines in the source code:

```
#define         SCS_ID1_K    0xFC3A
#define         SCS_ID2_K    0x69D3
```

These are used to provide a validation token at the beginning of each saved block of data in the FLASH. If you do not change these values, the CPU will read the existing FLASH and finding a match to the above keys will assume those saved parameters are valid. (Including the locked-out bit). So, in order to 'clear' the FLASH, you must change the tokens above. Then when the CPU starts it will find a difference in the FLASH values and assume the FLASH is invalid.

LED Blink Patterns

The on-board LED will blink out patterns to inform the user of its current status, errors, and pending actions (e.g., about to restart). Patterns are made up by a combination of blink patterns, and the speed at which they blink. The following table describes the patterns.

Status	Blink Pattern
Idle	
Ramp Bulk	
Accept	
Over Charge	
Float Post-float	
Equalize	
Error	Pattern repeated twice then followed by flashing out of error # (2 or 3 digits.)
Restarting	

If using the CAN enabled Systems Regulator, the LED will blink GREEN during normal operation. In addition, if the regulator is linked into a 'system' and being coordinated by a remote battery manager, the LED will blink YELLOW instead of Green. These colors can be used to quickly identify who device is currently acting as the battery master and that the system is configured correctly. It also allows quick visualization of which node takes over in the case of a failed battery master. The LED will blink RED if there is a fault condition.

APPENDIX A: RECEIVING DATA FROM THE REGULATOR:

- AST: Alternator STatus- sent every 1 second
- SST: System STatus -
- CST: CAN STatus – 3rd generation regulator only
- CPE:n Charge Profile Entry – line 'n' from the Charge Profile Table
- SCV: System Configuration Variables –
- NPC: Name / Password Configuration (was BTC: BlueTooth Configuration)
- FLT: System is FAULTED, Fault number will follow.
- AOK: Acknowledge – Regulator has received a valid change command.
- DBG: DeBuG – Additional internal information sent if the #define DEBUG is present.
- RST: ReSeTing – Regulator is resetting in response to a command or resettable fault.

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. Not the presence of dbl commas between major 'sections', this is to easy 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.

ALTERNATOR STATUS -- AST;

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 Alternator Regulator are directly reported as *AltVolts* and *AltAmps*. Unless overridden by an external source (example, the $EOR ASCII command) those same values will be assumed to be *BatVolts* and *BatAmps* and used b the regulator to make charge state decisions. See XXDFSF for more detailed description of the relationship of Alt and Bat volts/Amps and how they are utilized.

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:

0,1	– Alternator Off
2, 3	- Alternator FAULTED (See Fault Code)
4	- Alternator in delay mode while engine warms up
5	- Ramping towards BULK mode.
6,7	- In BULK mode
8	- In ACCEPTANCE mode
9	- In OVER CHARGE mode
10	- In FLOAT mode
11	- In FORCED_FLOAT mode (via Feature_in pin and CPE = #8)
12	- In OFF (Post Float) mode
13	- In EQUALIZE mode
14	- In CVCC mode (only available in system under direction of CAN master)

BTemp: Measured temperature of NTC sensor attached to B-port in degrees F 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 F. -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 defrees F.
-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 Alternator Regulator is configured with no limits for Alternator Amps and/or System Watts, the Alternator Regulator will self impost 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.*

SYSTEM STATUS -- SST;

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

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 (1-4x) 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 this last charge cycle.

WHs: The number of What Hours produced in this last charge cycle.

CAN STATUS -- CST;

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

(This string is only send out by the CAN enabled regulators)

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 my 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 disables, 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 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 Alternator Regulator as the RBM typically would occur only with small installations, example where the entire coordinated charging system consists of perhaps two engines and their alternators. However, one is also able to configuration more extensive system where the 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. Will = 0 if the Alternator Regulator has not associated itself with a remote master.

IgnoringRBM?: 0 or 1, Is the regulator ignoring the Remote Battery Master? (1 = Yes)
If the Remote Battery Master sends information which seems unbelievable the Alternator Regulator will set this flag and 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 Alternator Regulator has been assigned.

CHARGE PROFILE ENTRY – CPE;

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, 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 alternator 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 preformed. *floatVBAT* and *floatAMPS* will however be regulated to .

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 F (normalized to ‘12v’ battery)

CompMin: Minimum temperate to apply compensation at. In degree’s F

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.

SYSTEM CONFIG – SCV;

"SCV;, Lockout, fav32V?, RevAmp, SvOvr, BcOvr, CpOvr, ,AltTempSet, drtNORM, drtSMALL, drtHALF, PBF, ,Amp Limit, Watt Limit, , Alt Poles, Drive Ratio, Shunt Ratio, ,IdleRPM, TachMinField:"

Lockout:	Current lockout level. (0..2), see $SCO: command.
fav32V?:	0 or 1: Favor 32v system detection over 24/48v? (1 = yes)
RevAmp:	0 or 1: Reverse polarity of Amp Shunt readings? (1 = yes)
SvOvr:	System Voltage auto-detect (=0), or force (1..4 → 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 F
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 of TACH MODE is enabled. (IdleRPM and TachMinField: Firmware version 1.0.0 and above)

NAME & PASSWORD CONFIG – NPC;

"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)

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

FAULTED -- FLT;

FLT: “FLT;, FaultCode”

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

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

ACKNOWLEDGE – AOK;

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

DEBUG – DBG;

Special string with extra internal parameters. See Source code for details of Debug String

RESET – RST;

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:

- $RCP: n -Request to be send back CPE entry #N (n=1..8)
- $RCP:0 - Request to be sent back current selected CPE
- $RSS: - Request System Status be sent back
- $RSC: - Request System Configuration be sent back
- $RCS: - Request CAN Configuration be sent back
- $RNP: - Request Name & Password (Bluetooth/CAN) Configuration be sent back

- $CPA:n - Change ACCEPT parameters in CPE user entry n (n = 7 .. 8)
- $CPO:n - Change OVERCHARGE parameters in CPE user entry n (n = 7 .. 8)
- $CPF:n - Change FLOAT parameters in CPE user entry n (n = 7 .. 8)
- $CPP:n - Change POST-FLOAT parameters in CPE user entry n (n = 7 .. 8)
- $CPE:n - Change EQUALIZE parameters in CPE user entry n (n = 7 .. 8)
- $CPB:n - Change BATTERY parameters in CPE user entry n (n = 7 .. 8)
- $CPR:n - RESTORES Charge Profile n to default (as defined at program compile time).
  System will reboot after this command.

- $SCA: - Changes ALTERNATOR parameters in System Configuration table
- $SCT: - Changes TACHOMETER parameters in System Configuration table
- $SCN: - Changes NAME parameters in Configuration table
- $SCO: - Override: system voltage auto detect and/or DIP Switches for Charge profile & Battery Capacity,
  Lockout..
- $SCR: - RESTORES System Configuration table to default (as defined at program compile time).
  System will reboot after this command.

- $CCN: - Change parameters in the CAN Configuration table
- $CCR: - RESTORES CAN Configuration table to default (as defined at program compile time).
  System will reboot after this command

- $EBA: - External Battery Amps, allows external coordination of many charging sources.
- $EDB: - Enables DeBug strings. Has like effect to un-commenting “#define DEBUG” in source code.
- $FRM: - Force Regulator Mode. Forces regulator into charge step, ala into Equalize.
- $MSR: - RESTORE all parameters: $SCR:, $CCR: table, calibration, and every one of the $CPR: profiles.
  System will reboot after this command.
- $RBT: - ReBooT: Useful after a configuration changes have been made, to restart.

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 ‘Regulator Name & Password: need to Initialize’ on page# 31

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.

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 Smart Alternator Regulator which are not supported via standard RV-C DGNs. See AltReg_CAN.ccp for more details.

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

This command will instruct the Smart 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 sent 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 Smart 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 Smart 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.

$RSS: -Request to send back SST **

This command will instruct the Smart Alternator Regulator to send out via the Serial port a copy of the SST; System Status ASCII string.

$RSS:

$RSC: -Request to send back SSC**

This command will instruct the Smart Alternator Regulator to send out via the Serial port a copy of the SSC; System Configuration ASCII string.

$RSC:

$RCS: -Request to send back CST **

This command will instruct the Smart Alternator Regulator to send out via the Serial port a copy of the CST; CAN Status ASCII string.

$RCS:

$RNP: -Request to send back Name and Password **

This command will instruct the Smart Alternator Regulator to send out via the Serial port a copy of the NPC; Name & Password Configuration ASCII string.

$RNP:

$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 sytem (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).

Finally, If the regulator is unable to measure Amps, it will default to Adaptive Acceptance mode as well in order to provide protection to the battery. There is no way (outside of modifying the source code) to defeat this protection capability; instead make sure the Smart Alternator Regulator is able to measure Amps.

Note that in the case of two regulators connected to the same battery and sharing a regulator-synchronization cable, the sum of the local Amps and the remote Amps being delivered into the battery will be used to judge against this value. AMP based monitoring will be used.

CAUTION: If you set BOTH '*ExitDuration'* & *'ExitAmps'* = 0, then the regulator will never leave ACCEPT mode until the power is removed.

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 sytem (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 until the power is removed, of *'RevertAmps'* are exceeded.

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 then 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 *‘VBatSetPoint’* 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 they fall to (or below) *‘ExitAmps’* equalization will be terminated and the regulator will move to FLOAT mode. Not 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 porton 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}F>, < 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 F increments. This value is a floating-point number and entered for a normalized 12v system (See note: Defining Charging Voltages and Amps). Set = 0.0 to disable temperature based voltage compensation.

Min Com Temp: <WHOLE NUMBER (-20 → 100)> Additional compensation to battery target voltages will be stopped when the battery is at or below this temperature.

Min Charge Temp: <WHOLE NUMBER (-50 → 20) > If the battery drops below this temperature, the system will be forced into FLOAT mode to protect it.

Max Charge Temp: <WHOLE NUMBER (70 → 200) > 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 eventual FAULT based on the value of #define FAULT_BAT_TEMP in the source code (140f by default).

$CPR:n - RESTORES Charge Profile 'n' to default (as defined at program compile time).

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.

$SCA: - Changes ALTERNATOR parameters in System Configuration table

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

$SCA: <Favor 32v?>, < 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>

Favor 32v?: <WHOLE NUMBER (0, or 1) > There is too much cross over between expected voltages in a 32v system and a 24v systems , as well as a small amount of cross over with 48v systems. As such the regulator needs clarity when there is ambiguity. Setting this = 1 will instruct the regulator to 'favor' 32v system when there is a question. Setting this = 0 (default) will allow the regulator to favor 48v (and 24v) systems if there is any questions. If you have a 36v system, set this value = 1, else leave it = 0.

Another way to accomplish configuration of system voltage is to declare it directly via the $SCO: command (see *BC_index*). In the end, positively declaring the system voltage has higher reliability, but if you wish to retain some level of 'auto sampling', then *Favor32v*? flag can be used.

Alt Target Temp: <WHOLE NUMBER (60 → 240)> 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 alternators 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 $SCO command)
- Half Power Mode – Selected by shorting the Alternator NTC temperature sensor wires.

In operation, Derating 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 Alternator Regulator is able to determine PRMs (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 determine how quickly this pull-back is scaled off.

Set = 0 to disable this feature.

Set = -1 to cause Field Drive to be reduced to a maximum of 70% drive in the case where the 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 Alternator Regulator was able to measure RPMs successfully.

For many engine / alternator combinations the default value of 2 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 you 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 Alternator Regulator is enabled but the engine is not actually running, field drive fill 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 Status sensing wire being connected to the alternator. If the Alternator 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. It is used to protect the alternator from over current usage. There is no adjustment made to this value based on system voltage or selection of system battery size – the values declared in Amps 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.

Alt Amp Cap is a feature is used in Alternator-Centric deployments (see section '*Example Installations'* above). During operation the regulator will limit measured Amps to *Alt Amp Cap* value. During reduced power modes (ala – Half Power mode) the Amperage allowed will be reduced by the appropriate scaling factor. In this way of operation, sdfds asdfddfs WHAT ASD????).

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.

System Watts Cap: <WHOLE NUMBER (-1→ 20000)> This regulator will limit the system wattage to this value. Its primary use os 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). 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.

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 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 Alternator Regulator to increases its pull back of Field Drive during low RPM operations.

Set = 0 to enable 'auto' determination of Idle RPM.
(Firmware version 1.0.0 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 regulators name (Needed to unlock regulator, see "*Regulator Name & Password: need to Initialize*" on page 31)
- Disable Idle adaptive pullback (alternator has massive cooling capability, and engine sufficient reserve)
- Adjust amp shunt to 250A/75mV
- Define engine idle @ 550 RPMs
- Leave other values as default (See *Appendix D:)*

```
$SCN:0,MainsAlt,5555@
$SCA:0,200,1.0,0.75,0.50,0,0,0,3333,0,550@
$RBT:@
```

Notes: The Alternator Regulator is locked out from changes until the name/password has been updated at least once. Remember to turn on ALL the DIP switches with doing the 1st $SCN: command. 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 , 6hr max.
- 13.2v float - revert back if 2% of capacity is removed
- 15.3v Equalize, 3Hr duration.
- No overcharge nor post-float phases.
- 16.8mV temp comp

```
$SCN:0,MainsAlt,5555@
$SCA:0,200,1.0,0.75,0.50,0,0,0,3333,0,550@
$CPA:7 14.4,360,5,0@
$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 0.0168,15,-50,125@
$SCO:7,3,1,0@
$RBT:@
```

$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 Alternator Regulator.

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

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 Smart 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 in effect 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), the Regulator will not begin to apply field current until it is able to stably see RPMs, indicating the engine is running. Caution: in this situation, if the stator wire is not connected (or has failed), 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 value. Take note of this change and make adjustment during future use.

$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. Also allows the selection of auto detect for system voltage (12v, 24v, 32v, 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.

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 DIP switch functionality)

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. 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. No other command, not even $MSR: will be able to clear a non-zero lockout.

0 = No locking out.

1 = Prevent any configuration changes

2 = Prevent any configuration changes + ignore $EOR: inputs.

$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 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 capacaity:

- If not already done, set regulators name (Needed to unlock regulator, see “*Regulator Name & Password: need to Initialize*” on page 31)
- 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) parameters in System Configuration table

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 the alternator, the 1st time this 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 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 Bluetooth name and password are changed.
2. The Bluetooth name and password can only be initially changed if the Alternator is not charging, and all the DIP switch#1 is OFF, and DIP switched 2..8 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, or even power off the Bluetooth module using DIP switch position #1.

$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 2nd 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?>

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 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 Alternator Regulator attempt to act as the Remote Battery master?

0 = Do not allow the Alternator Regulator to assume an RBM status.
1 = the Alternator Regulator to assume an RBM status. (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 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 Alternator Regulator to only broadcase 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 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 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, the Alternator Regulator will stop sending out its own versions of this PGN. 0=no, 1=yes(default)

$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 2nd 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.

$EBA: - External Battery Amps

This allows for an external device to coordinate all charging sources and energy consumers within a system. It is used by the regulator to help decide when to change from Acceptance Phase or Overcharge phase into Float phase, as well as when to exit Equalize mode in cases where the Amp Shunt is not connected to the Battery, but the Alternator instead. (See section: EXAMPLE INSTALLATIONS)

$EAO: <Batt Amps>

Batt Amps: <FLOAT (-500.0 → 500.0)> Externally supplied battery amps. This will override the Amps Measured via the Amp Shunt when deciding to change Charge Modes from: Acceptance, Overcharge, Float, and Equalize. This value is held internal for 60 seconds before being forced = 0.0

$EBA: will reply with "AOK;" if the command was processed successfully, but will not be recognized if system is locked-out (see $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 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 Bluetoth 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.

$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 your 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>

---

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

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 length 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 command is accepted, "AOK;" will be returned. Note that even if the regulator is forced into a mode, it may quickly be taken out of that mode 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. (However, 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 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 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 Alternator Regulator over the CAN bus. This list may change, refer to the source code for more details.

NMEA-2000 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.
*/
```

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 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
     {127506L,NULL,                  &N2kDCStatus_message,      667,false},
     {127508L,&N2kDCBatStatus_handler,&N2kDCBatStatus_message,  667,false},
     {127513L,NULL,                  &N2kBatConf_message,         0,false},
     #endif
     #ifdef SUPPORT_RVC
     {0x1FFFD,&RVCDCStatus1_handler, &RVCDCStatus1_message,     500,false},
     {0x1FFFD, NULL,                 &RVCDCStatus1OA_message,   100,false},
     {0x1FFFC,&RVCDCStatus2_handler, &RVCDCStatus2_message,     500,false},
     {0x1FEC9,&RVCDCStatus4_handler, &RVCDCStatus4_message,    5000,false},
     {0x1FEC8,&RVCDCStatus5_handler, &RVCDCStatus5_message,     500,false},
     {0x1FEC8, NULL,                 &RVCDCStatus5OV_message,   100,false},         // Special instance, called every 100mS when over voltage.
     {0x1FEC7,&RVCDCStatus6_handler,           NULL,              0,false},
     {0x1FED0,&RVCDCDisconnectStatus_handler,  NULL,              0,false},
     {0x1FECF,&RVCDCDisconnectCommand_handler, NULL,              0,false},
     {0x1FFC7,&RVCChrgStat_handler,  &RVCChrgStat_message,     5000,false},
     {0x1FF9D,&RVCChrgStat2_handler, &RVCChrgStat2_message,     500,false},         /*  PROPOSED!!!  USING TEMP PGN# */
     {0x1FFC6,&sendNAK_handler,      &RVCChrgConfig_message,      0,false},         // For now we do not allow CPE configuration via the RVC protocol.
     {0x1FF96,&sendNAK_handler,      &RVCChrgConfig2_message,     0,false},
     {0x1FECC,&sendNAK_handler,      &RVCChrgConfig3_message,     0,false},
     {0x1FEBF,&sendNAK_handler,      &RVCChrgConfig4_message,     0,false},
     {0x1FF99, NULL,                 &RVCChrgEqualStat_message,5000,false},
     {0x1FF98,&sendNAK_handler,      &RVCChrgEqualConfig_message, 0,false},
     {0x17E00,&RVCTerminal_handler,  &RVCTerminal_message,       50,false},        // Terminal handler called 50mS to send 'Next portion' of string.
     #endif
         // J1939 type messages we need to handle.
     {0x1FECA, NULL,                 &ISODiagnostics_message,  5000,false},
     {0x1FECA, NULL,                 &ISODiagnosticsER_message,1000,false},        // Special instance, sent out more often during fault condition.

     {0,NULL,NULL,0,false}                                                         // ----PGN of 0 indicates end of table----
    };
```

Table 4: 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_1F_COMP;                         // Battery Temperature is compensated by this factor for every 1F temp change.  Note this
                                                            is based off of BAT_TEMP_NOMINAL (77f)
int            MIN_TEMP_COMP_LIMIT;                      // If battery temperature falls below this value (in deg-F), 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-f), 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-f),  stop charging and force into Float Mode to
                                                               protect it from over-temperature damage.

} CPS;
```

Appendix E: Default System Configuration

The following documents default values (As Compiled) for the Smart 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,                // .FAVOR_32V                  --> Do NOT favor 32v systems over 24/48v during auto-detection.
        false,                // .REVERSED_SHUNT             --> Assume shunt is not reversed.
        200,                  // .ALT_TEMP_SETPOINT          --> Default Alternator temp - 200f
        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 polls 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.