Programmable Voltage Reference Command Set and Calibration for Revision 3.0 firmware

Serial interface set up: The PVR has an USB connector on the Teensy 3.2 that can be used for communication along with a serial communications header that interfaces with an isolated USB interface board. The protocol is standard UART at 115200-8-N-1. That is, 115200 baud, 8 data bits, no parity and one stop bit. All messages to and from the PVR are readable ASCII text. The messages can be sent or received / monitored by a terminal emulator program, however no line editing (backspace, etc.) is supported.

There is a 30 second delay at turn on to allow voltages and the reference to stabilize. The default mode at turn on is Corrected mode using calibration offsets, and an output voltage of 2.500 volts.
When the PVR powers up it sends out the message "Programmable Voltage Reference V3.0" followed by a carriage return character.
Anytime the PVR reference voltage is manually changed using the rotary encoder, button, toggle switch, or command the PVR sends out a serial communications message containing the new Set reference voltage value, Offset value, and measured Output voltage value, followed by a carriage return character.

Message format: All commands sent to the PVR should be in the form "CNNNN<nl>" (do not include the quotes) where:

C is a single character indicating the command to be done. Valid values are '#', '!', 'W', 'R', 'U','N','C','O'. NNNN is four characters defining a decimal number. Each N is a digit character ('0' thru '9').

Together these form a number. The valid range for the number is defined in the context of the command.

<nl> (newline) is a single termination character. Valid values for <nl> are the carriage return character (0x0D) or the linefeed character (0x0A). If your terminal emulator sends both the extra will be ignored.

Command validation: If the message is less than six characters the message will be discarded. Beyond six characters will be ignored. If the message does not begin with one of the defined command characters the message will be discarded. If the message does not end with one of the <nl> characters the message will be discarded.

Note: Validation is **NOT** done on the N characters. Decimal conversion is done assuming each N is a valid digit character. If non-digit characters are given the decimal value will be generated based on their ASCII value.

Commands:

#: Sets the PVR reference value to NNNN millivolts, plus the internally stored calibration offset. The calibration offset value is displayed in the lower left Offset area of the OLED display. Only the bottom 13 bits of the reference value are used to program the DAC. A value of zero can be set with this command though the PVR may not actually output exactly zero volts. Values greater than 5000 will be treated as 5000. Front panel OLED displays COR for (corrected). The rotary dial LED is blue in this mode, and changes to violet if the output exceeds the error threshold. Default threshold is 5 mV.

!: Sets the PVR reference value to NNNN millivolts, without adding a calibration offset. Only the bottom 13 bits of the reference value are used to program the DAC. A value of zero can be set with this command though the PVR may not actually output exactly zero volts. Values greater than 5000 will be treated as 5000. The front panel OLED displays RAW, and the rotary dial LED is red in this mode.

W: Writes a value to onboard EEPROM memory. The previous set PVR reference value is used as the EEPROM address. The value to be written is created from NNNN. Only the lower 8 bits are used, therefore NNNN should be restricted to the range "0000" to "0255".
W0001<nl> would store an offset of +76.2 uV and W0255 would store an offset of -76.2 uV.

R: Returns the table of calibration offsets as a series of strings. Each string represents one onboard EEPROM memory location starting from zero. (**Not Fully Implemented**)
What I would like to implement is sending out a serial communications message with the address and offset value, starting at the current reference set value, and then send NNNN sequential address and offset pairs, and remain in the original mode when complete.

U: Sets an immediate global offset value to be used with the current reference value instead of using the offset from the calibration EEPROM memory. Similar to the W command, but isn't stored in EEPROM, and is lost on reset or loss of power. Front panel OLED displays UOS in this mode and the rotary dial LED is green. The set global offset is displayed in the lower left Offset area of the OLED display.
For example sending U0002<nl> would set a global offset of +152.4 mV. Any changes to the set reference value using the front panel controls would include the global offset.

N: Returns Normal behavior from U command to look up offsets from the calibration memory and clear the global offset. For example sending N0000<nl> would perform the Return to normal behavior.

O: Turns off the OLED display while leaving the PVR and internal reference powered up. A valid # or ! command restores the OLED display. An example of the OLED off command is O0000<nl> .
Valid range for NNNN is 0000 to 5000 and the reference output is set to the value entered.

C: Front panel Calibration mode. **Currently not implemented**.
The original intent of the C (Calibrate) function is to put the PVR in a mode that allows the use of the front panel switch and rotary encoder to easily change the calibration offset using a connected calibrated meter on the output.

Instead of typing in offsets thru the USB port, the offset would be changed using the +/- 25 mV switch in 76.2 uV increments with real time changes shown on the calibrated meter, entered and stored using the rotary push-button, and then move on to the next setting using the rotary dial.

A nice addition would be to automatically use the last offset setting for the next output setting, and then just press the push-button to save if the output is within cal limits.

Typically the output stays within cal for many steps before needing to be changed, and then it typically only needs to change by one cal step.

Front Panel:

The front panel includes the 2 line by 16 character OLED display. The top line of the display shows the set reference voltage, the measured output voltage, mode, and charging indicator. The lower line of the display shows the calibration or global offset value, firmware revision, and battery status.

The power button turns on and off all power the unit including battery charging.

The mode button currently only resets the mode and set reference voltage to the default state.

The toggle switch increases or decreases the set reference voltage by 25 mV.

The rotary dial / pushbutton / LED, adjusts the set reference voltage up (clockwise) or down (counterclockwise) in 1 mV steps, pressing the dial pushbutton increases the set value by 250 mV per press until 5.000 volts is reached, and then cycles to 0.001 volts, as does the +/- 25 mV switch, and the rotary dial. The dial LED indicates the current mode and an output error condition.

The +/- reference output banana jacks accept both 4mm standard and safety sheathed banana plugs, and use a standard 0.75" spacing between jacks. The set reference output voltage is adjustable between 0.001 to 5.000 volts in 1 mV steps with a maximum output current of 15 mA.

Back Panel:

The back panel includes a 7.5 volt DC input barrel power jack which needs a minimum 5 watt supply.

The isolated micro USB connector is also located on the back panel and requires 5 volt power thru the connector to operate, which is typical for most USB cables.

Firmware:

The firmware was coded in Arduino 1.8.X and Teensyduino 1.56.
There are several variables in the firmware that set the initial defaults which may be changed:

refVal = 2500   This sets the initial set reference voltage output to 2.500 volts.

errorvoltage = 5.0   Maximum error difference between set and output voltage in mV.

oled.setBrightness(50) This sets the initial OLED display screen brightness. Range 0 to 255

delay(30000)   Initial 30 second delay on startup, minimum 5 seconds (5000)

Push button and toggle switch default values of 250 mV and 25 mV.

Additional to-do items:

Add **A**uto-calibration routine for calibration of Teensy ADC read back of actual output voltage.