# *DISCLAIMER*

There is not a convention on the write-ability of registers. You may write to any register in the system including registers THAT MAY DAMAGE OR DESTROY your Classic and attached equipment like battery banks. Please make sure that you understand what you are doing before attempting to change any settings (like battery voltage) using the raw MODBUS interface. MidNite Solar cannot take responsibility for any damage to your Classic or system in the event of mis-configuration.

Registers and bits marked RESERVED are not necessarily unimplemented. Great care must be taken not to overwrite these registers or bits to ensure proper operation of the Classic. Some bits are marked to stay a 1 rather than 0.

Wherever possible we have tried to indicate settings that may have an adverse effect on your system if set incorrectly.

## *Conventions:*

Register units are expressed using formulas to try to reduce the ambiguity surrounding converting from the raw bus formats and human-readable values. Due to the nature of the Classic's operation, there is not a uniform convention as far as data endianness so you may find some classes of values that follow a Most Significant Byte (MSB) first convention and others that use the Least Significant Byte (LSB) first convention.

Bits are numbered from 0-15 in the little-endian or LSB-0 notation. That is, the least significant bit is indexed by 0.
For Example, the number 43,981 is 0xABCD in hex is represented in binary as 1010101111001101.

The bit ordering is as shown in the following table:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |

So that the binary digit indexed by (0) is 1. (1) is 0, (6) is 1, and (10) is 0.

MODBUS registers are 16-bit (2-octet/byte) in size. When using values from the map to indicate conversion formulas the following convention is used to access different octets:

Note that when talking about bytes or octets, it is more convenient to describe them in hexadecimal (base 16) form than in decimal. Some values are expressed in bytes as it is a more compact way of dealing with certain classes of numbers (IP addresses, for example). We use the 'C' convention of referring to hexadecimal numbers by prefixing them with '0x'. So the decimal value 10 is represented by the hexadecimal value 0x0A.

A note on Read/Write: The Classic MODBUS does not strictly enforce the Read/Write accessibility of some registers. It is possible to write to some registers marked Read Only. This will usually have no effect on Classic operation, but in some cases you may corrupt a register from which you wanted to read data. In some cases the registers are reset internally as new values. Additionally, some counters are accessible directly. For example, the Absorb Time (4139) timer is a "suggested" read only register; however, if your MODBUS application has reason to reset or adjust this register, there is nothing to prevent you from changing it however you please. Don't be surprised if your batteries pop in and out of Absorb stage unexpectedly, though!

Operators:

**Indexing**

[ ] square brackets around an address indicate the value of the register specified:

ex: [4116] indicates the value of the register at address 4116.

$[\ ]_{MSB}$ Square brackets followed by an $_{MSB}$ means to use the most-significant byte of the register.

ex: if the value at register 4116 is 0x04B1 (decimal 1201):

[4116] = 0x04B1,

then

$[4116]_{MSB}$ = 0x04

$[\ ]_{LSB}$ Square brackets followed by an $_{LSB}$ means to use the least-significant byte of the register.

ex: if the value at register 4116 is 0x04B1 (decimal 1201):

[4116] = 0x04B1,

then

$[4116]_{LSB}$ = 0xB1

( ) Parentheses refer to bits within the register.

For example: [4116](0) means "the 0th bit of the value in register 4116.

So if 4116 held the value 1 (0x0001 hex) then [4116](0) would be a 1. [4116](1…15) would all be 0s.

… Ellipses represent ranges of values or indices.

For example to refer to the first three bits of a register you may see:

[4116](0…3) which signifies the first three bits of the value of register 4116.

You may also see spans of registers as:

[4116…4120] which refer to registers 4116 to 4120 inclusive.

**Arithmetic**

\+ Add two numbers together

\- Subtract two numbers (or negate the value on the right)

/ Divide two numbers (integer implied)

* Multiply two numbers (integer implied)

## Note on addresses vs. registers:

The modbus specification adds one (1) to the "address" sent to the unit in the packet command to access a "register". This is so that modbus registers start at 1 rather than 0. The main Classic address map starts at register 4101 but the packet itself sends out address 4100.

Some modbus software and libraries will go by register number and some will go by address so make sure which one it works with.

**Logical**

<< x Binary shift left. Shift the value x binary digits to the left. This is equivalent to multiplying the decimal number by $2^x$. Ex:

[4116] = 0x01 = $00000001_b$

Then

[4116] << 1 = 0x02 = $00000010_b$

>> x Binary shift right. Shift the value x binary digits to the right. This is equivalent to dividing the decimal number by $2^x$.

[4116] = 0x02 = $00000010_b$

Then

[4116] >> 1 = 0x01 = $00000001_b$

| OR two numbers together (aligned to LSB)

& AND two numbers together (aligned to LSB)

^ XOR two numbers together (aligned to LSB)

**String**

|| Concatenate.

[4116] = 0x4142.

$[4116]_{MSB}$ || $[4116]_{LSB}$ => 0x41 || 0x42 => 'A' ||'B' => "AB"

**Examples:**

Using the example of the Average PV Voltage register: dispavgVpv @ 4116

Suppose using a MODBUS scanner you retrieve the following (integer) value from the dispavgVpv register at address 4116:

**4116**: 1201

- The address itself: 4116
- Full 16-bit value at the address: [4116] = 1201 (0x04B1 hex)
- The top-most (MSB) octet of the register: $[4116]_{MSB}$ = 0x04 hex
- The bottom-most (LSB) octet of the register $[4116]_{LSB}$ = 0xB1 hex
- Applying the Conversion (**[4116] /10) Volts:**
  - [4116] = 1201
  - 1201 /10 = 120.1 Volts

**File Transfer and Function Execution modes:**

MODBUS File Transfer and Function execution commands. File Transfer will be necessary, for instance, to transfer wind power curves to/from the Classic or logging and audio data to/from the Classic and/or MNGP. The standard Modbus file transfer commands 0x14 and 0x15 are a bit too "heavy" for embedded systems so Midnite uses our own internal commands 0x68 (read) and 0x69 (write) protocol, described at the end of this document.

**(EE)** When you see (EE), this means that register value is saved to EEprom whenever the Force write to EEprom is set and sent to the Classic.

When write to EEprom is requested, ALL registers that can be saved to EEprom are saved at this time.

| **Base Registers** *(registers ADDRESS plus 1)* | | | | |
|---|---|---|---|---|
| **Register** | **R/W** | **Name** | **Conversion** | **Notes** |
| 4101 | R | UNIT_ID | PCB revision = $[4101]_{MSB}$<br>Unit Type = $[4101]_{LSB}$ | The PCB revision is a value between 0 and 255 indicating the hardware revision of the PC board. The Unit Type is an integer value indicating the voltage category of the Classic See Table 4101-1. |
| 4102 4103 | R | UNIT_SW_DATE_RO | Year = [4102]<br>Month = $[4103]_{MSB}$<br>Day = $[4103]_{LSB}$ | Software Build date. |
| 4104 | R | **InfoFlagsBits3** | Information Flags word 3 | WhizBangJrCrcGood Bit 0x0001 |
| 4105 | RESERVED | | | |
| 4106 4107 4108 | R | UNIT_MAC_AddressI | $[4108]_{MSB}$ : $[4108]_{LSB}$ :<br>$[4107]_{MSB}$ : $[4107]_{LSB}$ :<br>$[4106]_{MSB}$ : $[4106]_{LSB}$ | The unit's Ethernet MAC address. |
| 4109 4110 | R | WbJrAmpHourNET (Clone of 4369) | (([4110] << 16) + [4109]) Amp Hours | Whizbang Jr. Net amp hours. (temporary) |
| 4111 4112 | R | UNIT_Device_ID | ([4112] << 16) + [4111] | The device ID of the unit. |
| 4113 | R | StatusRoll | ([4113]>>12)Count + ([4113]& 0x0fff) Value | Various 12 bit values changes once per second. Hi 4 bits = count |
| 4114 | R | RestartTimerms | [4114] Milliseconds | Time after which the Classic can wake up. (countdown) |
| 4115 | R | dispavgVbatt | ([4115] /10) Volts | Average Battery Voltage (1 sec) |
| 4116 | R | dispavgVpv | ([4116] /10) Volts | Average PV terminal input Voltage (1 second average) |
| 4117 | R | IbattDisplayS | ([4117] /10) Amps | Average Battery Current (1 sec) |
| 4118 | R | kW--Hours | ([4118] /10) kWatt-Hours | Average Energy to the Battery This is reset once per day |
| 4119 | R | Watts | [4119] Watts | Average Power to the Battery |
| 4120 | R | ComboChargeStage | Charge Stage = $[4120]_{MSB}$<br>State = $[4120]_{LSB}$ | See Table 4120-1 for battery charge state. **See Table 4120-2** |
| 4121 | R | PvInputCurrent | ([4121] /10) Amps | Average PV terminal input Current. (1 second average) |
| 4122 | R | VocLastMeasured | ([4122] /10) Volts | Last measured open-circuit Voltage at the PV terminal input. |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4123 | R | HighestVinputLog (EE) | [4123] Voltage /10 | Highest input voltage seen |
| 4124 | R | MatchPointShadow | [4124] Present wind power curve step being indexed (1…16) | |
| 4125 | R | AmpHours | [4125] Amp Hours | Daily Amp Hours reset at 23:59 |
| 4126 4127 | R | Lifetime kW-Hours (EE) | (([4127] << 16) + [4126]) kWh | Lifetime Energy Generation |
| 4128 4129 | R | LifetimeAmpHours (EE) | (([4129] << 16) + [4128]) Amp Hours | Lifetime Amp-Hour Generation |
| 4130<br>4131 | R | InfoFlagsBits<br>(InfoFlagsBits2) | ([4131] << 16) + [4130] | See Table 4130-1<br>(read as 32 bits or singly) |
| 4132 | R | BATTtemperature | ([4132] /10) °C | Temperature measured at the external Battery Temperature Sensor (if installed, else 25C) |
| 4133 | R | FETtemperature | ([4133] /10) °C | Temperature of Power FETs |
| 4134 | R | PCBTemperature | ([4134] /10) °C | Temperature of the Classic Control (top) PCB |
| 4135 | R | NiteMinutesNoPwr | [4135] minutes | Counts up when no power<br>Resets to 0 when there is power |
| 4136 | R/W | MinuteLogIntervalSec (EE) | [4136] seconds | Minimum 60 seconds recent history data logging interval |
| 4137 | R/W | modbus_port_register (EE) | [4137] | 0 to 65,535. Default = 502 |
| 4138 | R | FloatTimeTodaySeconds | [4138] seconds | Number of seconds that the Batteries have spent in float today. Reset at 23:59 (midnight) |
| 4139 | R/W | AbsorbTime | [4139] seconds | Absorb Time Up/Down Counter<br>Goes to Float when it reaches 0 |
| 4140 | **RESERVED** (may show Ibatt in older firmware) | | | |
| 4141 | R | PWM_ReadOnly | [4141] ( 0 to 1023) | Duty Cycle command of PWM signal. (NOT Percent) |
| 4142 | R | Reason_For_Reset | Reason the Classic reset (WDT, etc) | See Table 4142-1 |
| 4143 | R/W | Equalize Time | [4143] Seconds | Battery Stage Equalize Down Counter. Time remaining in Equalize Stage. EQ Done when it reaches 0 |
| 4144 | **RESERVED -- DO NOT WRITE** (was used for Solar Tracking debug) | | | |
| 4145 | R/*W* | MppWLast | [4145] Internal watts ref. for Solar Re-Tracking (~10% of last MPP W) If written to, will be overwritten next Solar sweep. | |
| 4146 | R/W | USBcommMODE (EE) | [4146] USB Function # | **See table 4146-1** |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4147 | R/W | NoDoubleClickTimer | [4142] Seconds | Internal forced time space between manual MPPT sweeps. |
| 4148 | R/W | Battery output Current Limit (EE) | [4148] /10) Amps | Battery Current Limit Amps (eg. 23.4 A = 234) |
| 4149 | R/W | Absorb Set Point Voltage (EE) | ([4149] /10) Volts | Battery Absorb Stage Set point Voltage (eg. 28.3V = 283) |
| 4150 | R/W | Float Voltage Set Point (EE) | ( [4150] /10) Volts | Battery Float Stage Set Point Voltage |
| 4151 | R/W | Equalize Voltage Set Point (EE) | ([4151] /10) Volts | Battery Equalize Stage Set Point Voltage |
| 4152 | R | Sliding Current Limit | [4152] Amps | Sliding Current Limit (changes with V/Temp etc.) |
| 4153 | R/W | Minimum Absorb Time (EE) | [4153] seconds (normally unused now) | Minimum Absorb time when VariMax is used. Otherwise, equals 0 |
| 4154 | R/W | Absorb Time (EE) | [4154] seconds | SetPoint time for Batteries to be in the Absorb Stage. |
| 4155 | R/W | Maximum Battery Temperature Compensation Voltage (EE) | ([4155] /10) Volts | Highest Charge Voltage is limited to this value when using battery temp sensor |
| 4156 | R/W | Minimum Battery Temperature Compensation Voltage (EE) | ([4155] /10) Volts | Lowest Charge Voltage is limited to this value when using battery temp sensor |
| 4157 | R/W | Battery Temp Comp Value for each 2V cell (EE) | -([4157] /10) mV/degree C/cell (0.5 mV steps) 0 to 10 mV per 2V cell | Absolute value of the Temperature Compensation Value in mV/°C /2V cell |
| 4158 | R/W | General Purpose 16 bit word stored & retrieved with other EEprom (was Battery Type) (EE) | | |
| 4159 | R/W | EqualizeReTryDays (EE) | [4159] Number of days for auto EQ to retry until giving up | |
| 4160<br>4161 | W | Force Flag Bits | ([4161] << 16) + [4160] | See **Table 4160-1**. |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4162 | R/W | Equalize Time (EE) | [4162] Seconds | Initialize Time for Batteries to remain in Equalize stage. |
| 4163 | R/W | Equalize Interval Days (EE) | [4163] Days | Number of days between Equalize Stages (Auto EQ) |
| 4164 | R/W | Mppt Mode (Solar, Wind, etc) (EE) | [4164] (bit 0 = On/Off) | Classic functional Mode. See Table 4164-1. |
| 4165 | R/W | Aux 1 and 2 Function (EE) | [4165] | Combined Aux 1&2 Functions + On/Off. See Table 4165-1 and 4165-2 |
| 4166 | R/W | Aux1VoltsLoAbs (EE) | ([4166] /10) Volts | Aux 1 Low Absolute Threshold Voltage |
| 4167 | R/W | Aux1DelayT (EE) | [4167] Milliseconds | Aux 1 Delay time before Asserting. |
| 4168 | R/W | Aux1HoldT (EE) | [4168] Milliseconds | Aux 1 Hold time before De-asserting. |
| 4169 | R/W | Aux2PwmVwidth (EE) | ([4169] /10) Volts | Voltage range over which PWM operates for Aux 2. 0,1,2,3,4 or 5 volts |
| 4170 | **RESERVED** | | | |
| 4171 | | | | |
| 4172 | R/W | Aux1VoltsHiAbs (EE) | ([4172] /10) Volts | Aux 1 High Absolute Threshold Voltage |
| 4173 | R/W | Aux2VoltsHiAbs (EE) | ([4173] /10) Volts | Aux 2 High Absolute Threshold Voltage |
| 4174 | R/W | Aux1VoltsLoRel (Relative to charge stage set point V) (EE) | ([4174] /10) Volts | Aux 1 Waste-Not Relative Lower Voltage Threshold (Charge Stage Relative V) |
| 4175 | R/W | Aux1VoltsHiRel (Relative to charge stage set point V) (EE) | ([4175] /10) Volts | Aux 1 Waste-Not Relative Upper Voltage Threshold (Charge Stage Relative V) |
| 4176 | R/W | Aux2VoltsLoRel (Relative to charge stage set point V) (EE) | ([4176] /10) Volts | Aux 2 Waste-Not Relative Lower Voltage Threshold (Charge Stage Relative V) |
| 4177 | R/W | Aux2VoltsHiRel (Relative to charge stage set point V) (EE) | ([4177] /10) Volts | Aux 2 Waste-Not Relative Upper Voltage Threshold (Charge Stage Relative V) |
| 4178 | R/W | Aux1VoltsLoPv (absolute) (EE) | ([4178] /10) Volts | Aux 1 Lower PV Absolute Threshold Voltage |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4179 | R/W | Aux1VoltsHiPv (absolute) (EE) | ([4179] /10) Volts | Aux 1 High PV Absolute Threshold Voltage |
| 4180 | R/W | VariMax [4180]$_{LSB}$ =Amps, [4180]$_{MSB}$ = Vabsorb –Vrelative /10 (Default = 101 amps) (EE) | | |
| 4181 | R/W | Aux2VoltsHiPv (absolute) (EE) | ([4181] /10) Volts | Aux 2 High PV Absolute Threshold Voltage |
| 4182 | R/W | EnableFlags3 (EE) | [4182] binary | See Table **4182-1** |
| 4183 | R/W | ArcFaultSenstvty (EE)<br>Requires Classic reset | Time = [4183]<br>Sense = [4183]<br>Mode = [4183] <<16 | Arc Fault Protection sensitivity response adjustments |
| 4184 | **RESERVED (Do NOT Write)** | | | |
| 4185 | | | | |
| 4186 | R/W | EnableFlags2 (EE) | [4186] binary | See Table **4186-1** |
| 4187 | R/W | EnableFlagsBits (EE) | [4187] binary | See Table **4187-1** |
| 4188 | R/W | **RESERVED FACTORY CALIBRATION** (Do NOT Write) | | |
| 4189 | R/W | Vbatt_Offset (EE) | ([4189] /10) | Battery Voltage Offset Tweak (Range Limited) (Signed) |
| 4190 | R/W | Vpv_Offset (EE) | ([4190] /10) | Input Voltage Offset Tweak (Range Limited) (Signed) |
| 4191 | R | VpvTargetRd | ([4191] /10) Volts | Input Target (V regulation) Voltage (Usually Vmpp) |
| 4192 | R/W | VpvTargetWr (EE) | ([4192] /10) Volts Vpv Target command (VpvTargetCmdEn) | |
| 4193 | **RESERVED (Do NOT Write)** | | | |
| 4194 | | | | |
| 4195 | | | | |
| 4196 | | | | |
| 4197 | R/W | SweepIntervalSecs (EE) | [4197] Seconds | Legacy P&O, Hydro, Solar, U-Set Sweep Interval, Seconds (Forcing Sweep resets timer) |
| 4198 | R/W | MinSwpVoltage (EE) | ([4198] /10) | Minimum input voltage for Hydro MPPT mode sweep |
| 4199 | R/W | MaxInputCurrent (EE) | ([4199] /10) amps (dflt = 99A) | Maximum input current limit |
| 4200 | R/W | SweepDepth (EE) | [4200] watts % | Maximum % Legacy/Hydro mode will sweep as percent of present Mpp wattage |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4201 | **R/W** | NegativeCurrent Adj. (EE) | Factory calibration. Do not change unless instructed. | |
| 4202 | R | ClipperCmdVolts | ([4202] /10) Volts | Aux clipper reference varies w/stage and headroom |
| 4203 | R/W | WindNumberOfPoles (EE) | [4203] poles | Number of turbine alternator poles for RPM Calc. (unused) |
| 4204 | R/W | MppPercentVoc (EE) | [4204] 00 to 100 % | % of Voc for U-Set mode |
| 4205 | R/W | WindTableToUse (EE) | [4205] | **FUTURE** power curve select |
| 4206 | **RESERVED (Do NOT Write)** | | | |
| 4207 | R/W | LEDmode (EE) | [4207] | See **Table 4207-1** |
| 4208<br>4209 | **RESERVED (Do NOT Write)** | | | |
| 4210<br>4211<br>4212<br>4213 | R/W | ID name (EE) | $[4210]_{LSB} \| [4210]_{MSB} \|$<br>$[4211]_{LSB} \| [4211]_{MSB} \|$<br>$[4212]_{LSB} \| [4212]_{MSB} \|$<br>$[4213]_{LSB} \| [4213]_{MSB}$<br><br>End with 0 if less than 8 chars | Unit Name. 8 characters max. ASCII. Takes place of MODBUS Register in MNGP display if present. **Example:** "CLASSIC" = 0x4C43, 0x5341, 0x4953, 0x0043 |
| 4214<br>4215 | R | CTIME0 | ([4215] << 16) + [4214] (possibly atomic op) | Consolidated Time Registers See **Table 4214-1** |
| 4216<br>4217 | R | CTIME1 | ([4217] << 16) + [4216] (possibly atomic op) | Consolidated Time Registers See **Table 4216-1** |
| 4218 | R | CTIME2 | [4218] | Consolidated Time Registers See **Table 4218-1** |
| 4219 | **RESERVED (Do NOT Write)** | | | |
| 4220 | R/W | RemoteMenuMode | [4220] | Present Remote Menu sent from MNGP |
| 4221 | W | RemoteButtons | [4221] MNGP Buttons Pressed See **TABLE 4221-1** | |
| 4222 | **RESERVED (Do NOT Write)** | | | |
| 4223 | **RESERVED (Do NOT Write)** | | | |
| 4224 | R/W | PreVoc **(do not write)** | ([4224] /10) Volts | Input Voc before Relay |
| 4225 | **RESERVED (Do NOT Write)** | | | |
| 4226 | R/W | Aux2_A2D_D2A | [4226] 10 bits, right justified | Aux 2 A/D and D/A value |
| | | | | |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4227 -4230 | **RESERVED (Do NOT Write)** | | | |
| | | | | |
| 4231 | R | VocRD | ([4231] /10) Volts | Last $V_{OC\ reading}$ |
| | | | | |
| **Register** | **R/W** | **Name** | **Conversion** | **Notes** |
| 4232 | **RESERVED (Do NOT Write)** | | | |
| 4233 | | | | |
| 4234 | | | | |
| 4235 | | | | |
| 4236 | R/W | AbsorbTime (duplicate) | [4236] seconds | Absorb Time Counter |
| 4237 | R/W | AntiClickSenstvty (EE) | [4237] | Best Left Alone (varies) |
| 4238 | R/W | SiestaTime | [4238] seconds | Sleep timer (5 minutes max) |
| 4239 | R | SiestaAbortVocAdj | ([4239] /10) Volts | Volts above last Voc reading to abort Siesta. |
| 4240<br>4241 | R | flagsRD | ([4241] << 16) + [4240] | Intrnl Flags See **Table 4240-1** |
| 4242 | **RESERVED (Do NOT Write)** | | | |
| 4243 | | | | |
| 4244 | R | VbattRegSetPTmpComp | ([4244] /10) Volts | Temperature compensated battery regulation target voltage |
| 4245 | R/W | VbattNominal (EE) | [4245] 12 * 1 thru 10<br>(120 Max for 250 KS) | Nominal Battery bank voltage (i.e. 12V, 24V, etc) |
| 4246 | R/W | EndingAmps (EE) | ([4246] /10) Amps<br>(Default = 0.0 amps) | Goes to Float below this Batt current if Temp Comp'd Absorb voltage is held |
| 4247 | R/W | EndingSoc (EE) | SOC to end Absorb | Future SOC use |
| 4248 | | EndAmpSocMBaddress | Modbus address of Ibatt | Future SOC use |
| 4249 | R/W | RebulkVolts (EE) | ([4249] /10) Volts | Rebulks if battery drops below this for > 90 Seconds |
| 4250 | R | BattMonVolts | Battery Monitor Volts | Future SOC use |
| 4251 | R | BattMonSOC | Battery Monitor SOC | Future SOC use |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4252 | R | DaysBtwnBulkAbs (EE) | Days between Bulk/Absorb cycles Default = 0 | If 1, will Re-bulk every other day, etc. |
| 4253 | R | BattMonAHefficiency (EE) | Battery Monitor A-Hour eff. | Future SOC use |
| 4254 | R/W | DayLogCombCatIndex | [4254] (Category << 10) + (-Day Offset & 0x3ff)<br>See information for **Daily logs Table 4254-1** | |
| 4255 | R | LogValueRead | [4255] Requested single logging Cat. data point appears here | |
| 4256 | R/W | MinLogCombCatIndex | [4256] (Category << 10) + (-Data Point Offset & 0x3ff)<br>See **info. for Recent/Hourly/Minutely logs Table 4256-1** | |
| 4257 | R/W | RebulkTimerSec | [4257] seconds Set Point | Rebulk interval timer seconds Cleared if Vbatt >= Rebulk V in Float MPPT |
| 4258<br>4259<br>4260<br>4261<br>4262<br>4263 | **RESERVED (Do NOT Write)** | | | |
| 4264<br>4265 | R/W | Voc_Qualify_Timer_1ms | (([4265] << 16) + [4264]) msec | Timer (msec) qualifying time till turn on valid (not very useful) |
| 4266 | R | IpvMinus Raw PV negative current from A/D converter | [4266] amps (peak A) | 1 count = 0.1 amp |
| 4267 - 4270 | **RESERVED (Do NOT Write)** | | | |
| 4271 | R | RestartTimerms | [4271] Milliseconds | Count Down Time until Wake Allowed at < 500 ms |
| 4272 | R | Ibatt | ([4272] /10) Amps (peak A) | Battery Current, Unfiltered |
| 4273<br>4274 | **RESERVED (Do NOT Write)** | | | |
| 4275 | R | ReasonForResting | [4275] Reason number | Reason Classic went to Rest (See **Table 4275-1**) |
| 4276 | R | Output Vbatt | ([4376] /10) Volts (peak V) | Battery Voltage Unfiltered |
| 4277 | R | Input Vpv | ([4377] /10) Volts (peak V) | PV Voltage Unfiltered |
| 4278 | **RESERVED (Do NOT Write)** | | | |
| 4279 | **RESERVED (Do NOT Write)** | | | |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4280 | **RESERVED (Do NOT Write)** | | | |
| 4281 | | | | |
| 4282 | R | PkHoldVpvStamp | Internal variable for Solar MPPT | |
| 4283 | R | VpvTargetRd (temporary) | [4283] /10 Vpv input voltage target (MPP voltage) | |
| 4284 | R | SwpDeepTimeoutSec | Internal variable for Solar MPPT | |
| 4285 | **RESERVED (Do NOT Write)** | | | |
| 4286 | R/W | LowWatts (EE) | [4284] Classic will go to Resting when watts are less than this for > 90 seconds in NON-wind modes unless Insomnia is set | |
| 4287 | R/W | WindLowWatts (dflt = 50) (EE) | Below this watts for wind, looks for power window | |
| 4288 | R/W | WindWindowWattsRef (EE) | If wind power wiggles above this, keep running | |
| 4289 | R | WindowWattsRO | Delta watts below WindLowWatts that power is wiggling | |
| 4290 | R/W | WindTimeOutRef (EE) | If power is low for this many seconds, go to Resting (default 90) | |
| 4291 | R/W | WindTimeOut2Ref (EE) | Seconds, Reference default = 1800 (1/2 hour) | |
| 4292 | R/W | WindTimeOut | Seconds, Timer. If > WindTimeOutRef , goes to Resting | |
| 4293 | R/W | WindTimeOut2 | Seconds, Timer. If > WindTimeOut2Ref, goes to Resting | |
| 4294 | R | MinVpvTurnOn | Minimum input voltage required to come out of Resting | |
| 4295 | R | VpvB4TurnOff | Internal reference of Vpv when going to Resting | |
| 4296 | R/W | HydroSwpAmps10Time6 | Internal reference for hydro sweep speed | |
| 4297 | R/W | EndingAmpsTimerSec | Seconds, Timer for Ending Amps. 60 seconds reference | |
| 4298 | R/W | PkAmpsOverLimitHi (EE) | Factory calibration. **Leave as is** | |
| 4299 | R/W | PkAmpsOverLimitLo (EE) | Factory calibration. **Leave as is** | |
| 4300 | R/W | FactoryVbattOffset (EE) | Factory V battery offset calibration. Normally, leave alone. | |
| 4301 | R/W | WindPowerTableV +0 (EE) | WindPowerTableV (stp 1) << 8) + WindPowerTableV(stp0) | |
| 4302 | R/W | WindPowerTableV +1 (EE) | WindPowerTableV (stp 3) << 8) + WindPowerTableV(stp2) | |
| 4303 | R/W | WindPowerTableV +2 (EE) | WindPowerTableV (stp 5) << 8) + WindPowerTableV(stp4) | |
| 4304 | R/W | WindPowerTableV +3 (EE) | WindPowerTableV (stp7) << 8) + WindPowerTableV(stp6) | |
| 4305 | R/W | WindPowerTableV +4 (EE) | WindPowerTableV (stp 9) << 8) + WindPowerTableV(stp8) | |
| | | | | |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4306 | R/W | WindPowerTableV +5 (EE) | WindPowerTableV (stp11) << 8) + WindPowerTableV(stp10) | |
| 4307 | R/W | WindPowerTableV +6 (EE) | WindPowerTableV (stp13) << 8) + WindPowerTableV(stp12) | |
| 4308 | R/W | WindPowerTableV +7 (EE) | WindPowerTableV (stp15) << 8) + WindPowerTableV(stp14) | |
| 4309 | R/W | WindPowerTableI +0 (EE) | WindPowerTableI (stp 1) << 8) + WindPowerTableI (stp0) | |
| 4310 | R/W | WindPowerTableI +1 (EE) | WindPowerTableI (stp 3) << 8) + WindPowerTableI (stp2) | |
| 4311 | R/W | WindPowerTableI +2 (EE) | WindPowerTableI (stp 5) << 8) + WindPowerTableI (stp4) | |
| 4312 | R/W | WindPowerTableI +3 (EE) | WindPowerTableI (stp7) << 8) + WindPowerTableI (stp6) | |
| 4313 | R/W | WindPowerTableI +4 (EE) | WindPowerTableI (stp 9) << 8) + WindPowerTableI (stp8) | |
| 4314 | R/W | WindPowerTableI +5 (EE) | WindPowerTableI (stp11) << 8) + WindPowerTableI (stp10) | |
| 4315 | R/W | WindPowerTableI +6 (EE) | WindPowerTableI (stp13) << 8) + WindPowerTableI (stp12) | |
| 4316 | R/W | WindPowerTableI +7 (EE) | WindPowerTableI (stp15) << 8) + WindPowerTableI (stp14) | |
| 4317 | **RESERVED (Do NOT Write)** | | | |
| **Register** | **R/W** | **Name** | **Conversion** | **Notes** |
| 4318 | R/W | PkAmpsOverTrip (EE) | Factory calibration. **Leave as is** | |
| 4319 | R | mngp_revision | Preliminary | Also shows unit is connected |
| 4320 | R | mnlp_revision | Preliminary | Also shows unit is connected |
| 4321 - 4324 | **RESERVED** (Do NOT Write) | | | |
| 4325 | **RESERVED** | | | |
| 4326 | R/W | ClassicModbusAddr (EE) | 0 to 255 Classic Modbus Addr | Default address = 10 (ten) |
| 4327 | R/W | BatteryTempPassed | Follow-Me temp sensor value | Follow Me |
| 4328 | R | iFlagsRO Low | Used for Follow-Me. Reflects neighboring units' charge stage for charge coordination. See **Table 4328-1** for relevant bits | |
| 4329 | R | iFlagsRO High | | |
| 4330 | R/W | ModbusControl | | Follow Me |
| 4331 | R/W | ClassicFmePassedBits | | Follow Me |
| 4332 | R/W | WindSynchA | Wind power tracking amps | Follow Me |
| 4333 | R/W | WindSynchV | Wind power tracking volts | Follow Me |
| 4334 | R/W | FollowMePassRef (EE) | Follow Me enabled if > 0. Classic Ignores changes for FollowMePassRef times around the communications loop. Should be set to at least number of units in Follow Me loop. | |
| | | | | |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4335 to 4340 | | **RESERVED** | | |
| 4341 | R | DabtU32Debug01 | Data Abort info if Watch Dog reset occurs | |
| 4342 | R | DabtU32Debug02 | | |
| 4343 | R | DabtU32Debug03 | | |
| 4344 | R | DabtU32Debug04 | | |
| 4345 to 4351 | | **RESERVED** | | |
| 4352 | R | CRC | Network CRC 16 | Network code area CRC |
| 4353 | R | CRC | Application CRC 16 | Application code area CRC |
| 4354 | R/W | ClearLogsCat | Clears various logging values | **See 4354-1** |
| 4355 | R | ClearLogsCounter10ms | Timer for sending 2nd ClearLogsCat conmand before timeout | |
| 4356 | R/W | User Variable 02 | General purpose user variable | |
| 4357 | R | WizBangRxBufferTempSh1 | Raw Whizbang Junior buffer read 1 | |
| 4358 | R | WizBangRxBufferTempSh2 | Raw Whizbang Junior buffer read 2 | |
| 4359 | R | WizBangRxBufferTempSh3 | Raw Whizbang Junior buffer read 3 | |
| 4360 | R | WizBangRxBufferTempSh4 | Raw Whizbang Junior buffer read 4 | |
| 4361 | R/W | WbangJrCmdS (EE) | Whizbang Junior Command. Default 0x35 | |
| 4362 | R | WizBangJrRawCurrent | WB Jr. raw current (not scaled or rounded) | (signed) A/D counts |
| 4363 | R/W | WbJrNumeratorSS (EE) | Whizbang Jr. gain adjustment Default = 0 | (signed) gain adjust +- 32,767 |
| 4364 | | Reserved | | |
| 4365 4366 | R | WbJrAmpHourPOSitive | (([4366] << 16) + [4365]) Amp Hours Pos. | (unsigned) Positive Whizbang Jr. amp hours |
| 4367 4368 | R | WbJrAmpHourNEGative | (([4368] << 16) + [4367]) Amp Hours Neg. | (signed) Negative Whizbang Jr. amp hours |

| Register | R/W | Name | Conversion | Notes |
|---|---|---|---|---|
| 4369 4370 | R | WbJrAmpHourNET | (([4370] << 16) + [4369]) Amp Hours NET | (signed) Whizbang Jr. Net amp hours. |
| 4371 | R | WzBangJrCurrent32Signed | Whizbang Jr. amps (scaled and rounded) | signed +- 3,276.7 amps |
| 4372 | R | WizBangJrRawCrcAndtemp | (Raw CRC << 8) \| Temperature & 0xff + 50C | WB Jr. CRC, Temperature in C -50 C |
| 4373 4384 | | Reserved | Reserved | |
| | | | | |

**Table 4101-1 Device Type**

| Name | Value | Description |
|---|---|---|
| Classic150 | 150 | Classic 150 |
| Classic200 | 200 | Classic 200 |
| Classic250 | 250 | Classic 250 |
| Classic250 KS | 251 | Classic 250 with 120 V Battery bank capability (lower current than 250) |

**Table 4120-1 Battery Charge Stage (HIGH Byte of ComboChargeStage register)**

| Name | Value | Description |
|---|---|---|
| Resting | 0 | Off , No Power, Waiting for Power Source, Battery V over set point, etc. |
| Absorb | 3 | Regulating battery voltage at Equalize Set point |
| BulkMppt | 4 | Max Power Point Tracking until Absorb (Bulk Terminate) Voltage reached |
| Float | 5 | Battery is FULL and regulating battery voltage at Float Set point |
| FloatMppt | 6 | Max Power Point Tracking. Seeking Float set point Voltage |
| Equalize | 7 | Regulating battery voltage at Equalize Set point |
| HyperVoc | 10 | Input Voltage is above maximum Classic operating Voltage |
| EqMppt | 18 | Max Power Point Tracking. Seeking Equalize set point Voltage |

**Table 4120-2 Classic States (LOWER Byte of ComboChargeStage register)**

| Name | Value | Description |
|---|---|---|
| Internal Resting state 0 | 0 | Resting |
| Internal state 1,2 | 1,2 | Waking /Starting |
| Internal state 3,4,6 | 3,4,6 | MPPT or Regulating Voltage |

**Table 4130-1 Info Flag Bits: READ ONLY (can read single 16 bit Low or High words if wanted)**

| Flag | Value | Description |
|---|---|---|
| Classic Over Temperature | 0x00000001 | Classic Over Temperature if set |
| EEPROM error | 0x00000002 | Classic EEprom read/write found an error if set |
| SerialWriteLock | 0x00000004 | Ethernet modbus register change is locked (password) |
| Equalize In Progress | 0x00000008 | Equalize Charge stage Active if set |
| **RESERVED** | 0x00000010 | **RESERVED** |
| **RESERVED** | 0x00000020 | **RESERVED** |
| **RESERVED** | 0x00000040 | **RESERVED** |
| EQ MPPT | 0x00000080 | Battery V is less than EQ Voltage (EQ MPPT) |
| In V is Lower Than Out | 0x00000100 | Input Voltage (PV) is lower than Vbatt if set |
| Current Limit | 0x00000200 | User current limit or internal temperature current limt reached |
| HyperVoc | 0x00000400 | Hyper Voc PV input V is above maximum Classic input rating |
| **RESERVED** | 0x00000800 | **RESERVED** |
| **RESERVED** | 0x00001000 | **RESERVED** |
| Battery Temp Sensor Installed | 0x00002000 | Battery temperature sensor installed if set |
| Aux1 State On | 0x00004000 | Aux 1 ON (aux 1 connector has V or relay is closed) |
| Aux2 State On | 0x00008000 | Aux 2 ON (aux 2 connector has V present) |
| GroundFaultF | 0x00010000 | Ground Fault detected if set |
| OCP (Over Current Protect) | 0x00020000 | Hardware Over Current occurred (was DefCon4ErrF ) |
| ArcFaultF | 0x00040000 | Arc Fault occurred if set |
| NegBatCurrentF | 0x00080000 | Negative battery current if set (backfeed out of PV input) |
| **RESERVED** | 0x00100000 | **RESERVED** |
| XtraInfo2DsplayF | 0x00200000 | Extra info is available to display (Shading, etc) |
| PvPartialShadeF | 0x00400000 | Partial Shade detected during SOLAR sweep (if enabled) |
| WatchdogResetF | 0x00800000 | watchdog reset flag indicator (for debugging) |
| LowBatteryVF | 0x01000000 | (VERY low battery) Vbatt is lower than 8.0 Volts |
| StackumperF | 0x02000000 | Stack Jumper is NOT installed if set |
| EqDoneF | 0x04000000 | EQ Finished. Resets when mode changed or new day |
| TempCompShortedF | 0x08000000 | Indication of shorted Temp Comp if set (all fans should come on if this is true) |
| UN-LockJumperF | 0x10000000 | UN-Lock Jumper NOT installed if set (If Lock Jumper is installed, Ethernet write protect is bypassed) |
| XtraJumperF | 0x20000000 | Extra Jumper is NOT installed if set |
| InputShortedF | 0x40000000 | PV input terminals are less then 1.0 volt if set |
| **RESERVED** | 0x80000000 | RESERVED |

**Table 4160-1 ForceFlagsBits (Write Only) (*can write to low or hi 16 bits independently if wanted*)**

| Name | Value | Description |
|---|---|---|
| RESERVED | 0x00000001 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00000002 | **RESERVED (Do NOT Set to 1)** |
| ForceEEpromUpdateWriteF | 0x00000004 | Write all current settings to internal EEPROM |
| ForceEEpromInitReadF | 0x00000008 | Force read of EEprom (UnDo if a NV register changed and has not been EEprom Updated yet) |
| ForceResetInfoFlags | 0x00000010 | Force ALL Info Flags to zero when set to 1 (Will reset info flags to 0) (or course some may pop back to 1 after resetting if applicable) |
| ForceFloatF | 0x00000020 | Force battery charge stage to Float |
| ForceBulkF | 0x00000040 | Force new Bulk/Absorb charge stage and reset timers |
| ForceEqualizeF | 0x00000080 | Force new Equalize stage |
| ForceNiteF | 0x00000100 | Force a new day (saves daily logs, resets daily kW-H) |
| RESERVED | 0x00000200 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00000400 | **RESERVED (Do NOT Set to 1)** |
| ForceSweepF | 0x00000800 | Force a Sweep or re-track (MNGP ENTER does this) |
| RESERVED | 0x00001000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00002000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00004000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00008000 | **RESERVED (Do NOT Set to 1)** |
| ResetAeqCounts | 0x00010000 | Re-Load Auto EQ Counter & Retry days from settings |
| RESERVED | 0x00020000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00040000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00080000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00100000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00200000 | **RESERVED (Do NOT Set to 1)** |
| RESERVED | 0x00400000 | **RESERVED (Do NOT Set to 1)** |
| ForceResetFaultsF | 0x00800000 | Reset all of faults (ArcFault, GndFault etc.) |
| **RESERVED** | 0x01000000 | **RESERVED (Do NOT Set to 1)** |

**Table 4164-1 MPPT MODE NOTE:** Bit 0 is the ON/OFF (Enable/Disable) Table shows modes as ON Subtract One (1) if showing mode as OFF. Will revert to OFF if mode changed while running.

| | Value | Description |
|---|---|---|
| PV_Uset | 0x0001 | U-SET MPPT MODE (includes MPPT ENABLED (On) FLAG i.e. if 0x0000 MPPT mode is OFF) |
| DYNAMIC | 0x0003 | Slow Dynamic Solar Tracking (old Solar 1 O & P ) |
| WIND TRACK | 0x0005 | Wind Track Mode |
| **RESERVED** | 0x0007 | Reserved for future Wind Learn Mode |
| Legacy P&O | 0x0009 | Legacy P & O sweep mode |
| SOLAR | 0x000B | Fast SOLAR track (old PV Learn mode) |
| HYDRO | 0x000D | Micro Hydro mode (similar to Legacy P&O) |
| **RESERVED** | 0x000F | **RESERVED** |

**AUX 1 and 2 modes**

**Extracted and encoded as combined in Aux12Function**

**Table 4165-1 AUX 1 Off – Auto – On** *(Extracted/Encoded as Aux12Function bits 6,7)*

| Name | Value | Description |
|---|---|---|
| Aux 1 Off | 0 | Aux 1 output is forced OFF (0 Volts) |
| Aux 1 Auto | 1 | Aux 1 operates automatically as defined in Aux1Funtion |
| Aux 1 On | 2 | Aux 1 output is forced ON (~14 Volts) |
| Aux 1 Unimplemented | 3 | Unassigned at present |

Aux1OffAutoOn = (((Aux12Function & 0xc0) >> 6));

**Table 4165-2 AUX 2 Off – Auto – On** *(Extracted/Encoded as Aux12Function bits 14,15)*

| Name | Value | Description |
|---|---|---|
| Aux 2 Off | 0 | Aux 2 output is forced OFF (0 Volts) |
| Aux 2 Auto | 1 | Aux 2 operates automatically as defined in Aux2Funtion |
| Aux 2 On | 2 | Aux 2 output is forced ON (~14 Volts) |
| Aux 2 Unimplemented | 3 | Unassigned at present |

Aux2OffAutoOn = ((Aux12FunctionS & 0xc000) >> 14);

**Table 4165-3 AUX 1 Function** ***(Extracted/Encoded as Aux12Function bits 0-5)***

| Name | Value | Description |
|---|---|---|
| DIVERSION SLOW HIGH | 1 | Non-PWM On at Vbatt > Aux1VoltsHiAbs (after Delay time)<br>Off at Vbatt < Aux1VoltsLoAbs (after Hold time) |
| Low Battery Disconnect High | 2 | Same as DIVERSION SLOW HIGH but Active Low |
| Waste Not High | 3 | Non-PWM On at Vbatt > Aux1VoltsHiAbs<br>Off at Vbatt < Aux1VoltsLoAbs (Active High) |
| Waste Not Low | 4 | Non-PWM On at Vbatt > Aux1VoltsHiAbs<br>Off at Vbatt < Aux1VoltsLoAbs (Active Low) |
| **RESERVED** | 5 | Unimplemented |
| **RESERVED** | 6 | Unimplemented |
| PV ON HIGH<br>(Active High based on PV input) | 7 | Active High if Out Vin > Aux1VoltsHiPv (After Delay Time)<br>Aux 1 output goes off if Vin < Aux1VoltsLoPv after Hold Time<br>(High is 14V out or Relay On, Low is 0V out or Relay Off) |
| PV ON LOW<br>(Active Low based on PV input) | 8 | Active Low Out if Vin > Aux1VoltsHiPv after Delay Time<br>Aux 1 output goes on if Vin < Aux1VoltsLoPv after Hold Time<br>(Low is 0V out or Relay Off, High is 14V out or Relay On) |
| **RESERVED** | 9 | Unimplemented |
| **RESERVED** | 10 | Unimplemented |
| **RESERVED** | 11 | Unimplemented |
| **RESERVED** | 12 | Unimplemented |
| TOGGLE TEST | 13 | Output toggles once per second On-Off-On-Off automatically |
| NITE LIGHT HIGH | 14 | Active High On at Dawn, Off at Dusk |
| DAY LIGHT HIGH | 15 | Active High Off at Dawn, On at Dusk |
| WIND CLIPPER CONTROL<br>(Active High) | 16 | Output goes High when V input is greater than last input V that yielded Absorb, Float or EQ,<br>PLUS 5.0 volts of headroom |
| FLOAT HIGH | 17 | Output On when Floating, Off when not Float, after 2 Seconds |
| FLOAT LOW | 18 | Output Off when Floating, On when not Float, after 2 Seconds |
| VENT FAN HIGH | 19 | Output On when Vbatt > Aux1VoltsHiAbs after 1 second<br>Output Off when Vbatt < Aux1VoltsHiAbs after 30 seconds |
| VENT FAN LOW | 20 | Output Off when Vbatt > Aux1VoltsHiAbs after 1 second<br>Output On when Vbatt < Aux1VoltsHiAbs after 30 seconds |
| GFP TRIP HIGH | 21 | Output On for ~0.150 seconds when Ground Fault occurs |

Aux1Function = Aux12Function & 0x3f;

**Table 4165-4 AUX 2 Function** ***(Extracted/Encoded as Aux12Function bits 8-13)***

| Name | Value | Description |
|---|---|---|
| DIVERSION HIGH PWM | 0 | Digital Out PWM Battery Diversion (Active High) |
| DIVERSION LOW PWM | 1 | Digital Out PWM Battery Diversion (Active Low) |
| WASTE NOT HIGH | 2 | Digital Out PWM Relative to Charge Stage Voltage Threshold Diversion (Use It Or Lose It) (Active High) |
| WASTE NOT LOW | 3 | Digital Out PWM Relative to Charge Stage Voltage Threshold Diversion (Use It Or Lose It) (Active Low) |
| **RESERVED** | 4 | Unimplemented |
| **RESERVED** | 5 | Unimplemented |
| TOGGLE TEST | 6 | Out Once per second On-Off-On-Off automatic toggle |
| PV V ON HIGH | 7 | Active High output if Vin exceeds Aux2VoltsHiPv |
| PV V ON LOW | 8 | Active Low output if Vin exceeds Aux2VoltsHiPv |
| **RESERVED** | 9 | Unimplemented |
| WIND CLIPPER CONTROL | 10 | PWM output Active High when V input is above PV input V plus headroom voltage |
| NITE LIGHT HIGH | 11 | Active High On at Dawn, Off at Dusk |
| DAY LIGHT HIGH | 12 | Active High Off at Dawn, On at Dusk |
| FLOAT HIGH OUTPUT | 13 | Active High output when Float stage is reached |
| FLOAT LOW OUTPUT | 14 | Active Low output when Float stage is reached |
| Active HIGH (input) turn off | 15 | > 6.0 volts into Aux 2 forces Classic to resting (15V max) |
| Active LOW (input) turn off | 16 | < 2.0 volts into Aux 2 forces Classic to resting (0 V min) |
| Active HIGH (input) Float | 17 | > 6.0 volts into Aux 2 forces Classic to Float (15V max) |
| Whizbang Junior (WB Jr.) | 18 | Aux 2 commands WB Jr. and receives info from WB Jr. |

**NOTE:** When Aux 2 is used as Input, Maximum input is 15 V, Minimum V is 0 volts

Aux2Function = (Aux12FunctionS & 0x3f00) >> 8; (Digital/Analog Input/Output)

**TABLE 4182-1 EnableFlags3 R/W Binary. NOTE:** Requires Classic Reset before change effects.

When Classic receives a packet for an address other than itself (example, 10 and 255), it forwards the information out other ports if enabled. Follow-Me normally forwards from Downstream to Upstream and MNGP too Upstream (out the Classic's MASTER jack to its neighbor's SLAVE jack) so these are normally enabled when these bits are equal to 0

Classics will always treat address 255 as if it is for it as well as its set modbus address.

**TABLE 4182-1 EnableFlags3 R/W Binary (Default = 0) New Register January 2013**

| Name | Bit | Description |
|---|---|---|
| **RESERVED** | 0x0001 | **RESERVED (Do NOT Set to 1)** |
| Rem2DwnStrmBus | 0x0002 | Forward Remote (MNGP) port to Downstream Bus enabled if 1 |
| UpStrmBus2DwnStrmBus | 0x0004 | Forward Upstream Bus to Downstream Bus enabled if 1 |
| UpStrmBus2RemBus | 0x0008 | Forward Upstream Bus to remote (MNGP) Bus enabled if 1 |
| DwnStrmBus2RemBus | 0x0010 | Forward Downstream Bus to Remote (MNGP) Bus enabled if 1 |
| DwnStrm2UpStrmBus | 0x0020 | Forward Downstream Bus to Upstream Bus enabled if 0 |
| Rem2UpStrmBus | 0x0040 | Forward Remote (MNGP) Bus to Upstream Bus enabled if 0 |
| ClscAdvertizeDis | 0x0080 | TCP/IP Advertise on port 4626 every 10 seconds enabled if 0 |
| SMA Mode Enable | 0x0100 | Enables SMA mode which also changes modbus RS232 forwarding operation |
| JimComFix | 0x0200 | Normally 0. Works in conjunction with SMA Mode Enable **Leave as 0 unless instructed otherwise** |
| **RESERVED** | 0x0400 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x0800 | **RESERVED (Do NOT Set to 1)** |
| WBJrAmpsInvert | 0x1000 | When set to 1, inverts the polarity of Classic's interpretation of shunt current from Whizbang Jr. |
| WBJrAhBlkClrF | 0x2000 | When set to 1, Whizbang Jr. NET amp-hours will clear to zero next new Bulk/Absorb cycle |
| AFpower5VdisableF | 0x4000 | If set to 1, disables power to the arc fault PCB. Also used for Low Power modes at night |
| *AFpower5VdisableF (Continued)* | *0x4000* | This AF power down feature is available from approximately Serial number 5500 on 12-4-2012 |
| *AFpower5VdisableF (Continued)* | *0x4000* | Control board 8-003-1 Rev J1 and newer |
| FetTempEqulsPcbTmp | 0x8000 | **Leave at 0** unless otherwise instructed by MidNite Solar |

**Table 4186-1 EnableFlags2 bits [4186] NOTE: This is a newer Register and has had some bits transferred from Register EnableFlagsBits 4187 starting with revision date 12-1-2013**

| Name | Value | Description |
|---|---|---|
| DCclipperEn | 0x0001 | Aux 2 Clipper mode runs at lower frequency if 1 |
| **RESERVED** | 0x0002 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x0004 | **RESERVED (Do NOT Set to 1)** |
| PvPartialShadeEn | 0x0008 | Partial Shade reporting Enabled when Set to 1 |
| WasteNotPkRespF | 0x0010 | Aux 1 diverts according to Peak Vbatt if 1, Average Vbattery if 0 |
| BattTempNetEn | 0x0020 | DefCon3 Error reporting enabled if set to 1 |
| DefCon4ErrEn2 | 0x0040 | **(OCP) LEAVE SET ( 1 )** |
| PwmLowMaxFlag | 0x0080 | Low Max Enabled if Set to 1 (Low Vin - Vout) |
| VbatRegSlowFlag | 0x0100 | Average Vbatt Regulation if 1. Peak V if 0 Dflt = 0 |
| EndAmpWbJr | 0x0200 | If set to 1, Ending Amps uses Whizbang Junior Shunt. If 0, Classic shunt |
| BumpWindI | 0x0400 | When adjusting wind curve, automatically “bumps” adjacent current set points out of the way if set to 1 |
| DivrsnAbsEqTmrEn | 0x0800 | Enables Absorb & EQ timer counting when Aux1 or Aux2 functions are diverting battery V if set to 1 |
| StayOnWhileZeroEn (Insomnia) | 0x1000 | If set to 1, Insomnia is enabled and Classic will not Rest when 0 watts out |
| VpvTargetCmdEn | 0x2000 | User can adjust Target V in U-Set mode through VpvTargetWr Register when set to 1. |
| LogAtNiteEn | 0x4000 | Enable Hourly/Fast data logging while resting if 1 |
| WindSynchEnF | 0x8000 | Wind Synch Follow-Me enabled if 1 |

**Table 4187-1 EnableFlagsBits [4187] NOTE: Some EnableFlagsBits moved to EnableFlags2 in release version 12-1-2012 for Classic Lite and MNLP Compatibility**

| Name | Value | Description |
|---|---|---|
| GroundFaultEn | 0x0001 | Ground Fault Protection Enabled when Set to 1 |
| ArcFaultEn | 0x0002 | Arc Fault Protection Enabled when Set to 1 |
| **RESERVED** | 0x0004 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x0008 | MOVED to EnableFlags2 0x0008 |
| **RESERVED** | 0x0010 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x0020 | **RESERVED (Do NOT Set to 1)** |
| DefCon4ErrEn (OCP) | 0x0040 | Over Current Protection enabled (**LEAVE as 1**) |
| **RESERVED** | 0x0080 | MOVED to EnableFlags2 0x0080 |
| **RESERVED** | 0x0100 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x0200 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x0400 | MOVED to EnableFlags2 0x0400 |
| **RESERVED** | 0x0800 | MOVED to EnableFlags2 0x0800 |
| **RESERVED** | 0x1000 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x2000 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x4000 | **RESERVED (Do NOT Set to 1)** |
| **RESERVED** | 0x8000 | **RESERVED (Do NOT Set to 1)** |

**Table 4207-1 LED Mode**

| Name | Value | Description |
|---|---|---|
| All Off | 0 | No LED activity except on startup |
| Rick Mode | 1 | Minimal Activity: MNGP Yellow LED indicates Current Limiting or FET Temperature Limit if plugged into top jack |
| Blinky | 2 | All LEDs Active: Lightshow!<br>MNGP Red LED when MNGP->Classic MODBUS is active |
| LED 1 | 3 | Status Mode:<br>MNGP Green LED = Battery Full (Float)<br>Classic Red LED = Aux 1 Active.<br>Classic Yellow LED = Aux 2 Active<br>Classic Blue LED = Unimplemented |
| LED 2 | 4 | Comm. test. Flashes LED when modbus transaction |

**Table 4214-1 Consolidated Time Registers 0 (Read Only from Classic -- Normally, MNGP will Classic time from its battery backed RTC through file transfer)**

| Name | Value | Description |
|---|---|---|
| BITS 5:0 | 0 to 59 | Seconds Seconds value in the range of 0 to 59 |
| BITS 5:0 | RESERVED | **RESERVED** |
| BITS 13:8 | 0 to 59 | Minutes value in the range of 0 to 59 |
| BITS 15:14 | RESERVED | **RESERVED** |
| BITS 20:16 | 0 to 23 | Hours value in the range of 0 to 23 |
| BITS 23:21 | RESERVED | **RESERVED** |
| BITS 36:24 | 0 to 6 | Day Of Week Day of week value in the range of 0 to 6 |
| BITS 31:27 | RESERVED | **RESERVED** |

**Table 4216-1 Consolidated Time Registers 1 (Read Only from Classic – Normally, MNGP will Classic time from its battery backed RTC through file transfer)**

| Name | Value | Description |
|---|---|---|
| BITS 4:0 | 1 to 28, 29, 39, 31 | Day of month value in the range of 1 to 28, 29, 30, or 31(depending on the month and whether it is a leap year) |
| BITS 7:5 | RESERVED | **RESERVED** |
| BITS 11:8 | 1 to 12 | Month value in the range of 1 to 12 |
| BITS 15:12 | RESERVED | **RESERVED** |
| BITS 27:16 | 0 to 4095 | Year value in the range of 0 to 4095 |
| BITS 31:28 | RESERVED | **RESERVED** |

**Table 4218-1 Consolidated Time Register 2 (Read Only from Classic – Normally, MNGP will Classic time from its battery backed RTC through file transfer)**

| Name | Value | Description |
|---|---|---|
| BITS 11:0 | 1 to 366 * | Day of year value in the range of 1 to 365<br>* (366 for leap years) |
| BITS 31:12 | RESERVED | **RESERVED** |

**Table 4221-1 Remote (MNGP) Buttons Pressed NOTE:** MNGP buttons are sent to Classic when MNGP is in its Main status screen to change some parameters.

| Name | Value | Description |
|---|---|---|
| ENTER_key | 0x0010 | This key forces a sweep from MNGP's main status screen |
| MAINMENU_key | 0x0020 | |
| STATUSMENU_key | 0x0040 | |
| SOFTR_key | 0x0200 | Raises MPP V in modes except Wind Track from MNGP |
| SOFTL_key | 0x0400 | Lowers MPP V in modes except Wind Track |
| RIGHT_key | 0x0080 | |
| LEFT_key | 0x0100 | |
| DWN_key | 0x0800 | |
| UP_key | 0x1000 | |

**Table 4240-1 Internal Flags bits (Read Only) (flagsRD)**

| Name | Value | Description |
|---|---|---|
| **RESERVED** | 0x00000001 | **RESERVED** |
| **RESERVED** | 0x00000002 | **RESERVED** |
| **RESERVED** | 0x00000004 | **RESERVED** |
| **RESERVED** | 0x00000008 | **RESERVED** |
| **RESERVED** | 0x00000010 | **RESERVED** |
| **RESERVED** | 0x00000020 | **RESERVED** |
| **RESERVED** | 0x00000040 | **RESERVED** |
| AbsorbTimeRunf | 0x00000080 | Bulk/Absorb Timer Counting is Enabled |
| EqTimeRunf | 0x00000100 | EQualize Timer Run flag |
| FloatTimeRunf | 0x00000200 | Float Time accumulate flag |
| kWhAccumRunf | 0x00000400 | kiloWatt-hour & Amp-Hour accumulate enabled |
| **RESERVED** | 0x00000800 | **RESERVED** |
| AbsorbCountUpf | 0x00001000 | Absorb Timer is counting UP if set |
| OK2WriteIVtables | 0x00004000 | OK to write WindPowerTableV[] & I[] |
| WindLowflag | 0x00008000 | Indicates that we turned off because of low power |
| SweepDwnEnabledf | 0x00010000 | Slowly Sweeping Down V input |
| LowLightflag | 0x00020000 | State 2 detected low light (May not be accurate) |
| **RESERVED** | 0x00040000 | **RESERVED** |
| **RESERVED** | 0x00080000 | **RESERVED** |
| **RESERVED** | 0x00100000 | **RESERVED** |
| SweepUpEnabledf | 0x00200000 | Slowly Sweeping UP V input |
| **RESERVED** | 0x00400000 | **RESERVED** |
| **RESERVED** | 0x00800000 | **RESERVED** |
| **RESERVED** | 0x01000000 | **RESERVED** |
| BattFull | 0x02000000 | Battery Full, Absorb complete, Float |
| **RESERVED** | 0x04000000 | **RESERVED** |
| EqCountUpf | 0x08000000 | EQ Timer is counting Up |
| **RESERVED** | 0x10000000 | **RESERVED** |
| **RESERVED** | 0x20000000 | **RESERVED** |
| **RESERVED** | 0x40000000 | **RESERVED** |
| **RESERVED** | 0x80000000 | **RESERVED** |

**Table 4254-1 DayLogCombCatIndex (Daily Logs Combined Category and day Index)**

| | |
|---|---|
| Daily Log Category = (DayLogCombCatIndex & 0xfC00) >> 10 | Type of log value requested |
| Daily Log Index = DayLogCombCatIndex & 0x03ff | Days before today requested |

| Category | Value | Description |
|---|---|---|
| 0 | kW-Hours | kW-Hours /10 |
| 1 | Amp-Hours | Amp-Hours |
| 2 | Float Time | Float Time, Seconds |
| 3 | Time Stamp, Low | Month = TimeStampLow & 0x0F |
| | | Day of Month = (TimeStampLow & 0x01F0) >> 4 |
| | | Year = ((TimeStampLow & 0xFE00) >> 9) + 2000 |
| 4 | High Power, Watts | Battery Watts |
| 5 | High Temp, deg. C | FET Temperature, Degrees C /10 |
| 6 | Time Stamp, High | Minute = TimeStampHigh & 0x3F |
| | | Hour = (TimeStampHigh & 0x07C0) >> 6 |
| 7 | High V PV | Daily High PV Voltage /10 |
| 8 | High V Battery | Daily High Battery Voltage /10 |
| 9 | Low V Battery | Future Low Vbatt. Not implemented yet. |

**Table 4256-1 MinLogCombCatIndex (Minute Logs Combined Category and data point Index)**

| | |
|---|---|
| Minute Log Category = (MinLogCombCatIndex & 0xfC00) >> 10 | Type of log value requested |
| Minute Log Index = MinLogCombCatIndex & 0x03ff | Data point before now requested |

| Category | Value | Description |
|---|---|---|
| 0 | Output Power, Watts | |
| 1 | PV Input Voltage | Vpv /10 |
| 2 | Battery Voltage | Vbatt /10 |
| 3 | Time Stamp, Low | Month = TimeStampLow & 0x0F |
| | | Day of Month = (TimeStampLow & 0x01F0) >> 4 |
| | | Year = ((TimeStampLow & 0xFE00) >> 9) + 2000 |
| | | |
| 4 | Time Stamp, High | Minute = TimeStampHigh & 0x3F |
| | | Hour = (TimeStampHigh & 0x07C0) >> 6 |
| | | |
| 5 | Charge Stage | See **Table 4120-1** for charge stages at current time stamp |
| 6 | Output Current, Amps | Amps /10 |
| 7 | kW-Hours | kW-Hours /10 |

**LogValueRead** [4255] is updated with the value requested from **DayLogCombCatIndex** or **MinLogCombCatIndex**, whichever changed more recently.

**(Daily and Minutely/Hourly/Recent logs can also be read via file transfer… See end section, 1.0)**

**Table 4275-1 Reason For Resting**

| VALUE | REASON FOR RESTING |
|---|---|
| 1 | Anti-Click. Not enough power available (Wake Up) |
| 2 | Insane Ibatt Measurement (Wake Up) |
| 3 | Negative Current (load on PV input ?) (Wake Up) |
| 4 | PV Input Voltage lower than Battery V (Vreg state) |
| 5 | Too low of power out and Vbatt below set point for > 90 seconds |
| 6 | FET temperature too high (Cover is on maybe ?) |
| 7 | Ground Fault Detected |
| 8 | Arc Fault Detected |
| 9 | Too much negative current while operating (backfeed from battery out of PV input) |
| 10 | Battery is less than 8.0 Volts |
| 11 | PV input is available but V is rising too slowly. Low Light or bad connection (Solar mode) |
| 12 | Voc has gone down from last Voc or low light. Re-check (Solar mode) |
| 13 | Voc has gone up from last Voc enough to be suspicious. Re-check (Solar mode) |
| 14 | Same as 11 |
| 15 | Same as 12 |
| 16 | Mppt MODE is OFF (Usually because user turned it off) |
| 17 | PV input is higher than operation range (too high for 150V Classic) |
| 18 | PV input is higher than operation range (too high for 200V Classic) |
| 19 | PV input is higher than operation range (too high for 250V or 250KS) |
| 22 | Average Battery Voltage is too high above set point |
| 25 | Battery Voltage too high of Overshoot (small battery or bad cable ?) |
| 26 | Mode changed while running OR Vabsorb raised more than 10.0 Volts at once OR Nominal Vbatt changed by modbus command AND MpptMode was ON when changed |
| 27 | bridge center == 1023 (R132 might have been stuffed) This turns MPPT Mode to OFF |
| 28 | NOT Resting but RELAY is not engaged for some reason |
| 29 | ON/OFF stays off because WIND GRAPH is illegal (current step is set for > 100 amps) |
| 30 | PkAmpsOverLimit… Software detected too high of PEAK output current |
| 31 | AD1CH.IbattMinus > 900 Peak negative battery current > 90.0 amps (Classic 250) |
| 32 | Aux 2 input commanded Classic off. for HI or LO (Aux2Function == 15 or 16) |
| 33 | OCP in a mode other than Solar or PV-Uset |
| 34 | AD1CH.IbattMinus > 900 Peak negative battery current > 90.0 amps (Classic 150, 200) |
| 35 | Battery voltage is less than Low Battery Disconnect (LBD) Typically Vbatt is less than 8.5 volts |

**Table 4328-1 iFlagsRO for Follow Me:**

| R/W | BIT | Description |
|---|---|---|
| R | 0x00000040 | Equalize |
| R | 0x00000080 | Bulk/Absorb |
| R | 0x00000200 | Float |

The network registers are all Read/Write. You may write any values to these registers, however this may result in erratic operation in some instances.

To set A static IP address, be sure to clear the DHCP bit in register 20481 before writing the static values to the IP Address registers.

| **Network** | | | |
|---|---|---|---|
| **Register** | **Name** | **Units** | **Description** |
| 20481 | IP Settings | [20481] | Network Settings Flags. See Table 20481-1 |
| 20482<br>20483 | IP Address | $[20483]_{MSB}$ . $[20483]_{LSB}$ .<br>$[20482]_{MSB}$ . $[20482]_{LSB}$ | The IP address of the Classic† |
| 20484<br>20485 | Gateway Address | $[20485]_{MSB}$ . $[20485]_{LSB}$ .<br>$[20484]_{MSB}$ . $[20484]_{LSB}$ | Network Gateway Address.† |
| 20486<br>20487 | Subnet | $[20487]_{MSB}$ . $[20487]_{LSB}$ .<br>$[20486]_{MSB}$ . $[20486]_{LSB}$ | Network Subnet Mask† |
| 20488<br>20489 | DNS_1 | $[20489]_{MSB}$ . $[20489]_{LSB}$ .<br>$[20488]_{MSB}$ . $[20488]_{LSB}$ | Primary DNS Address† |
| 20490<br>20491 | DNS_2 | $[20491]_{MSB}$ . $[20491]_{LSB}$ .<br>$[20490]_{MSB}$ . $[20490]_{LSB}$ | Secondary/Alternate DNS Address† |
| 20492<br>20493 | **W** Serial Number<br>(Unlock Code)<br>For writing to Classic modbus registers over Ethernet | $[20492]_{MSB}$ . $[20493]_{LSB}$ | The serial number is read from 28673/28674 but it is written to the Ethernet configuration registers: 20492 and 20493. Write the Classic's serial number over Ethernet to unlock writing of modbus registers over Ethernet. Setting this will last until the TCP/IP connection is dropped<br><br>Example: the serial number is: 0x12345678 (hex)<br><br>20492 = MSB (Serial number) 0x1234<br>20493 = LSB (Serial number) 0x5678 |

† Read Only if the DHCP flag is set. To assign a static IP to the Classic, first clear the DHCP flag in the IP Settings Register (20481).

**See 4354-1 ClearLogsCat** Clears various logging values (Available June 2014)

To clear logs, send 0x8000 with LSB containing category of log to clear to Register 4354. Next, wait at least 750 milliseconds then send 0x4000 with LSB containing category of log to clear. If successful, Classic will return 0x0500

| Name | Value | Description |
|---|---|---|
| ClearLogsCat = 0x8000 \| ClrCat | | |
| Send to register 4354 | | |
| Wait 750 milliseconds | | |
| ClearLogsCat = 0x4000 \| ClrCat | | |
| | | |
| ClrCat | ClearLogsCat \|= 1 | Clears DAILY LOGS |
| ClrCat | ClearLogsCat \|= 2 | Clears HOURLY /MINUTELY LOGS |
| ClrCat | ClearLogsCat \|= 3 | Clears LIFETIME KW-HOURS |
| ClrCat | ClearLogsCat \|= 4 | Clears LIFETIME AMP-HOURS |
| ClrCat | ClearLogsCat \|= 5 | Clears Whizbang Junior NET AMP-HRS |
| ClrCat | ClearLogsCat \|= 6 | Clears Whizbang Junior POSITIVE AMP-HRS |
| ClrCat | ClearLogsCat \|= 7 | Clears Whizbang Junior NEGATIVE AMP-HRS |
| | | |
| Wait for result from register 4354… Operation may take a few seconds | | |
| | | |
| Success sends 0x0500 | | |
| Failure sends 0x0A00 | | |

**Table 20481-1 Network Settings Flags**

| Name | Value | Description |
|---|---|---|
| DHCP | 0x0001 | Set this bit to enable DHCP. |
| Web Access | 0x0002 | Set this bit to enable online access to your Classic through http://www.mymidnite.com |

The Version Registers are all Read Only. Writes to any of these registers will have no long-term effect.

| **Version** | | | |
|---|---|---|---|
| **Register** | **Name** | **units** | **Description** |
| 16385 | app_version | Major: [16385](15…12)<br>Minor: [16385](11…8)<br>Release: [16385](8..4) | Release version of the application code |
| 16386 | net_version, | Major: [16386](15…12)<br>Minor: [16386](11…8)<br>Release: [16386](8…4) | Release version of the communications stack |
| 16387<br>16388 | app_rev | ([16388] << 16) + [16387] | Build Revision of the application code |
| 16389<br>16390 | net_rev | ([16390] << 16) + [16389] | Build Revision of the communications code stack |
| 28673<br>28674 | **R** Classic serial number | ([28673] << 16) + [28674] Read Classic's serial number over RS-232<br><br>Lock jumper on Classic disables necessity of this for password over Ethernet. | |
| 20492 20493 | **W** Classic serial number | The serial number is read from 28673/28674 but it is written to the Ethernet configuration registers: 20492 and 20493. | |

Communication Statistics are all Read/Write registers. You may write any value to these registers which will be incremented should the trigger for that counter occur. The most useful type of write may be to periodically reset the counters to zero. These are all lifetime counters and due to the number of MODBUS transactions may overflow to 0.

| **Communication Statistics** | | | |
|---|---|---|---|
| **Register** | **Name** | **units** | **Description** |
| **Master / In Bus Interface** | | | |
| 10001<br>10002 | rx_ok | ([10002] << 16) + [10001] | Number of correctly received packets |
| 10003<br>10004 | rx_crc_err | ([10004] << 16) + [10003] | Number of packets received with crc errors |
| 10005<br>10006 | requested_ok | ([10006] << 16) + [10005] | Number of transactions originating from this unit that completed successfully |
| 10007<br>10008 | requested_err | ([10008] << 16) + [10007] | Number of transactions originating from this unit that failed |
| 10009<br>10010 | forwarded | ([10010] << 16) + [10009] | Number of packets forwarded through this interface |
| 10011<br>10012 | broadcast | ([10012] << 16) + [10011] | Number of broadcast packets received |
| 10013<br>10014 | dropped_busy | ([10014] << 16) + (10013) | Number of packets dropped due to the interface being busy. |
| 10015<br>10016 | **RESERVED** | | |
| **Slave / Out bus interface** | | | |
| 10017<br>10018 | rx_ok | ([10018] << 16) + [10007] | Number of correctly received packets |
| 10019<br>10020 | rx_crc_err | ([10020] << 16) + [10020] | Number of packets received with crc errors |
| 10021<br>10022 | requested_ok | ([10022] << 16) + [10021] | Number of transactions originating from this unit that completed successfully |
| 10023<br>10024 | requested_err | ([10024] << 16) + [10023] | Number of transactions originating from this unit that failed |
| 10025<br>10026 | forwarded | ([10026] << 16) + [10025] | Number of packets forwarded through this interface |
| 10027<br>10028 | broadcast | ([10028] << 16) + [10027] | Number of broadcast packets received |

| | | | |
|---|---|---|---|
| **Communication Statistics** | | | |
| 10029<br>10030 | dropped_busy | ([10030] << 16) + [10029] | Number of packets dropped due to the interface being busy. |
| 10031<br>10032 | **RESERVED** | | |
| **Remote bus interface** | | | |
| 10033<br>10034 | rx_ok | ([10034] << 16) + [10033] | Number of correctly received packets |
| 10035<br>10036 | rx_crc_err | ([10036] << 16) + [10035] | Number of packets received with crc errors |
| 10037<br>10038 | requested_ok | ([10038] << 16) + [10037] | Number of transactions originating from this unit that completed successfully |
| 10039<br>10040 | requested_err | ([10040] << 16) + [10039] | Number of transactions originating from this unit that failed |
| 10041<br>10042 | forwarded | ([10042] << 16) + [10041] | Number of packets forwarded through this interface |
| 10043<br>10044 | broadcast | ([10044] << 16) + [10043] | Number of broadcast packets received |
| 10045<br>10046 | dropped_busy | ([10046] << 16) + [10045] | Number of packets dropped due to the interface being busy. |
| 10047<br>10048 | **RESERVED** | | |
| **TCP bus interface** | | | |
| 10049<br>10050 | rx_ok | ([10050] << 16) + [10049] | Number of correctly received packets |
| 10051<br>10052 | rx_crc_err | ([10052] << 16) + [10051] | Number of packets received with crc errors |
| 10053<br>10054 | requested_ok | ([10054] << 16) + [10053] | Number of transactions originating from this unit that completed successfully |
| 10055<br>10056 | requested_err | ([10056] << 16) + [10055] | Number of transactions originating from this unit that failed |
| 10057<br>10058 | forwarded | ([10058] << 16) + [10057] | Number of packets forwarded through this interface |
| 10059<br>10060 | broadcast | ([10060] << 16) + [10059] | Number of broadcast packets received |

| **Communication Statistics** | | | |
|---|---|---|---|
| 10061<br>10062 | dropped_busy | ([10062] << 16) + [10061] | Number of packets dropped due to the interface being busy. |
| 10063<br>10064 | **RESERVED** | | |

| **Reserved** | | | |
|---|---|---|---|
| **Register** | **Name** | **units** | **Description** |
| 61441-61442 | Reserved | | Reserved |
| | | | |

# 1.0 File Transfer Introduction

The MidNite Solar Classic retrieves logs using file transfers. We use an internal or user-defined function to handle MODBUS file transfers. This is lighter-weight than the dedicated MODBUS file transfer commands using less bandwidth and leveraging a linear block addressing model which allows for internal flexibility. Unfortunately it means that users must write their own MODBUS handlers to generate and parse the data logs.

## *1.1 104 Read Internal Command*

| | |
|---|---|
| function code | 104 (0x68) |
| device | 8-bit (see Table 1.3-1) |
| data_length | 8-bit (maximum of 200) |
| reserved | 16-bit |
| address | 32-bit (see sections 1.3.1 and 1.3.2) |

The Read Internal Command accesses the Classic's file system in order to retrieve data from different devices onboard. To retrieve data logs, the device field is populated with one of the log types specified in Table 1.3-1. Depending on the log type selected the address field will then be populated by the specification for that particular log.

The data_length field specifies the number of bytes to be read from the specified device. In the case of log files, though, this field is ignored and the record is returned in its entirety.

The reserved field is ignored and should be set to 0.

### 1.1.2 Read Internal Response

| | |
|---|---|
| device | 8-bit (Memory, Logs, Set Time, etc) |
| data_length | 8-bit (maximum of 200) |
| reserved | 16-bit (reads as 00 00) |
| address | 32-bit |
| Data array returned | Size of data_length in bytes (octets) |

The Read Internal Response is the MODBUS response packet returned after successful execution of the Read Internal Command (1.1). The device, data_length, reserved, and address fields will be the same as in the Read Internal Command that solicits this response.

The data field is an octet array representing the type of data that was requested. This differs based on the device field. See section 1.3.1 and 1.3.2 for details on the data format for log transfers..

## *1.2 Command Details*

There are different devices available for reading on the Classic. This document only details the log file transfers however. Log files take one of two forms: daily logs which contain accumulated details for one day's worth of data and minutely logs which contain instantaneous values for data fields at the logging data rate. In order to retrieve one of these logs, the **device** field of the Read Internal Command should be set to one of the values in table 1.3-1.

**Table 1.3-1 Device Fields**

| Device Field Category | Device | Description |
|---|---|---|
| modbus_file_memory | 4 | Addrs 0 = WindTableV, Addrs 1 = WindTableI |
| modbus_file_dailies_log | 5 | Read Daily Logs from EEprom |
| modbus_file_minutes_log | 6 | Read Minutely Logs from EEprom |
| modbus_file_TimeDateRiseSet | 7 | Write Sunrise/Sunset --- Set Time/Date |

### 1.2.2 105 Write Internal Command

| | |
|---|---|
| function code | 105 (0x69) |
| device | 8-bit See Table 1.3-1 |
| data_length | 8-bit (maximum of 200) |
| reserved | 16-bit |
| address | 32-bit (see sections 1.3.1 and 1.3.2) |
| Data array to send | data_length octets |

### 1.2.3 Write Internal Response

| | |
|---|---|
| device | 8-bit |
| data_length | 8-bit (maximum of 200) |
| Reserved | 16 bit (reads as 00 00) |
| address | 32-bit |

(not including any TCP/IP header or CRC)

### 1.3.1 modbus_file_dailies_log

When reading from the data log, it is necessary to specify which particular field is required and from which day offset you would like to retrieve the data.

| Address | |
|---|---|
| bits 15..10 | bits 9..0 |
| category index | day index |

The category index refers to which particular data field (e.g. kWh) you would like to retrieve (see Table 1.3.1-1).

The Day Index represents the number of the day prior to today that you would like to retrieve. For instance, to retrieve today's data you would use index 0, yesterday's data would be index 1, the day before yesterday's would be 2, *etc.*

As an example, to retrieve yesterday's kWh total, one would assemble the address field as:

Category Index: 0 (Energy / kWh)
Day Index: 1 (yesterday)

Address = ((category index & 0x003F) << 10) + (day index & 0x03FF) = 0x00000001

where the 0x fields represent hexadecimal notation and the << operator represents a logical shift left operator (by 10 bits in this example.)

As another example, to retrieve time in float from this day last week:

Category Index = 2
Day Index = 7 (7 days ago)

Address = ((category index & 0x003F) << 10) + (day index & 0x03FF) = 0x00000807

**Table 1.3.1-1 (Also see Table 4254-1)**

| Daily Category Index | |
|---|---|
| 0 | Energy (kWh) |
| 1 | Amp Hours (Ah) |
| 2 | Time in Float (TBD) |
| 3 | Timestamp Low |
| 4 | High Power (W) |
| 5 | High FET temperature (C) |
| 6 | Timestamp High |
| 7 | Input Voltage (V) |
| 8 | Battery Voltage (V) |

**Timestamp Translation:**

Timestamps are encoded as follows:

| Timestamp Low | | | |
|---|---|---|---|
| bits 15 | bits 14..9 | bits 8..4 | bits 3..0 |
| - | Years after 2000 | Day of Month | Month |

| Timestamp High | | |
|---|---|---|
| bits 15..11 | bits 10..6 | bits 5..0 |
| - | Hours | Minutes |

## 1.3.2 modbus_file_minutes_log

When reading from the data log, it is necessary to specify which particular field is required and from which day offset you would like to retrieve the data.

| Address | |
|---|---|
| bits 15..10 | bits 9..0 |
| category index | sample index |

The category index refers to which particular data field (e.g. kWh) you would like to retrieve (see Table 1.3.2-1).

The Day Index represents the number of the sample prior to now that you would like to retrieve. For instance, to retrieve the most recent sample you would use index 0, the last sample would be 1.

**Table 1.3.2-1 (Also see Table 4256-1)**

| Minutely (Hourly/Recent History) Category Index | |
|---|---|
| 0 | Power (W) |
| 1 | Input Voltage (V) |
| 2 | Battery Voltage (V) |
| 3 | TimeStamp Low |
| 4 | TimeStamp High |
| 5 | Charge Stage Combo |
| 6 | Output Current (A) |
| 7 | Energy (kWh) |

Timestamps are decided as specified in section 1.3.1.

As an example, to retrieve the most recent Energy reading one would assemble the address field as:

Category Index: 7 (Energy / kWh)
Day Index: 0 (last reading)

Address = ((category index & 0x003F) << 10) + (day index & 0x03FF) = 0x00001C00

where the 0x fields represent hexadecimal notation and the << operator represents a logical shift left operator.

### 1.3.3 Wind Power Curve Tables (Device 4)

| Address | Array Contents | Description |
|---|---|---|
| 0 | WindPowerTableV [16] | 16 Bytes. Voltage steps starting at V Cut-in of turbine<br>0 to 255 volts e.g. [64,68,70,72,75,78,81,83,85,87,89,91,93,98,104,112] |
| 1 | WindPowerTableI [16] | 16 Bytes. Battery current, steps starting at 0 amps Cut-in<br>0 to 255 amps e.g. [0,2,4,6,8,10,15,20,25,30,35,40,45,50,55,60] |

#### 1.3.4 modbus_file_ SunriseSunsetTime[20] (Device 7) (Internal Clock stops while writing)

| | Array SunriseSunsetTime[] Contents | Description |
|---|---|---|
| | SunriseSunsetTime[0] through SunriseSunsetTime[7] | Unused but must be sent |
| | SEC = SunriseSunsetTime[11] & 0x3f; | Seconds |
| | MIN = SunriseSunsetTime[10] & 0x3f; | Minute |
| | HOUR = SunriseSunsetTime[9] & 0x1f; | Hour (0-23) |
| | DOW = SunriseSunsetTime[8] & 0x07; | Daw of Week |
| | DOM = SunriseSunsetTime[15] & 0x1f; | Day of Month |
| | MONTH = SunriseSunsetTime[14] & 0x0f | Month |
| | YEAR = (short)((SunriseSunsetTime[13] \|((SunriseSunsetTime[12]) << 8) )); | Year |
| | DOY = (SunriseSunsetTime[19] \|(SunriseSunsetTime[18] << 8)) & 0x0fff ; | Day of year |

Default RS-232 format is Baud Rate 19.2K, No parity, 8 bits, 1 stop bit (N81)

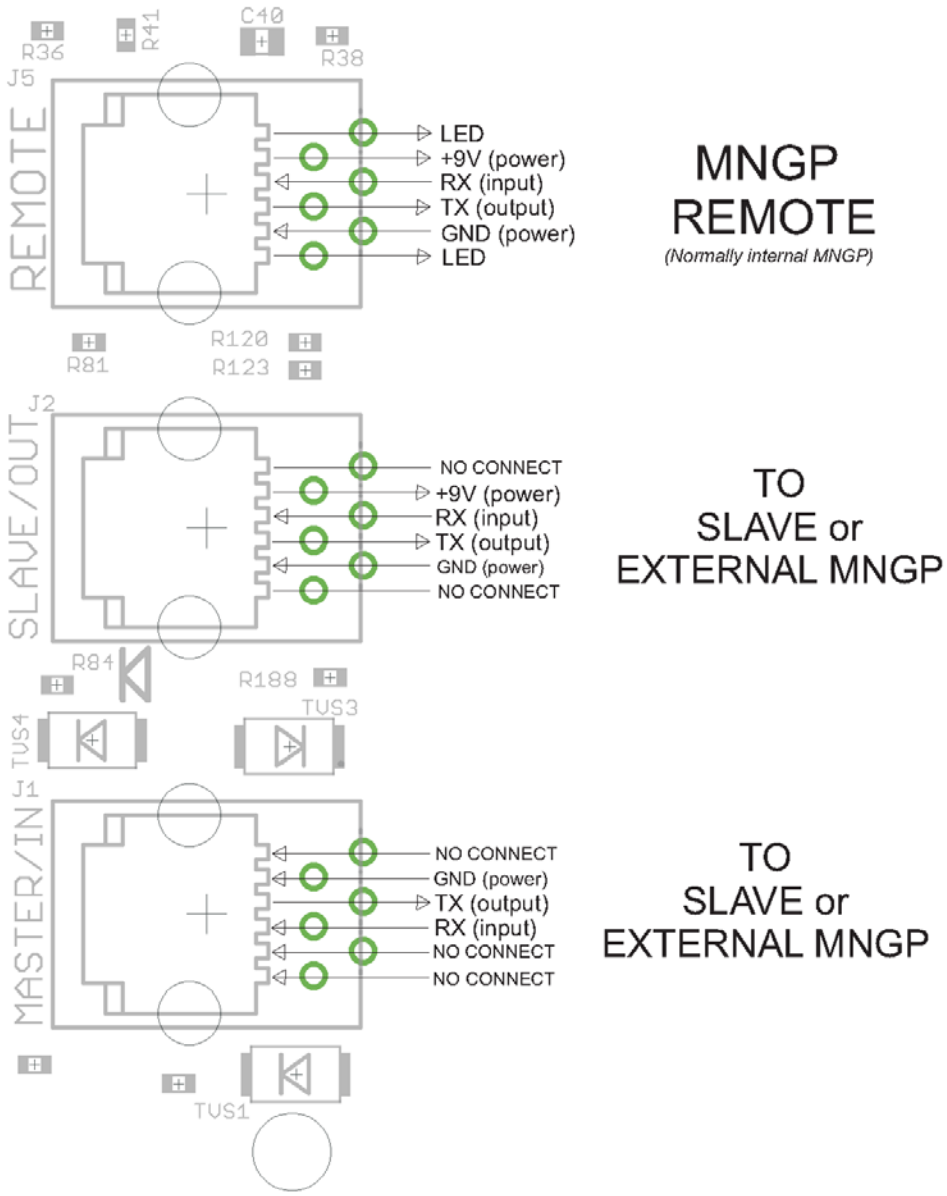


Figure 1. Classic RS-232 MODBUS RJ-11 PHONE JACK PINOUTS (Top View)