Maker Module – Hybrid Electric Water Heater (HEWH) Interface Definition

# Interface Modes:

The interface between the green bean and the HEWH can take place while the appliance is in either a Consumer or a Native mode. These modes are defined as follows:

**Consumer Mode**: Allows programmers to access appliance high-level algorithms available to the consumer. In **consumer mode** a user connected with the green bean can modify the operational modes or set temperature of the unit (or other functions available via the front panel), but can not change the low level functions that govern how the heating cycles run.

**Native Mode**: When a programmer uses the green bean to connect to the appliance in **native mode**: the API facilitates low-level, direct control of motors, fans, actuators, heaters, and other controlled devices.  High-level algorithms, such as a Hybrid Electric mode are not operational.  Native mode allows programmers to, for example, create a new water heating profile to meet unique needs.

# ⚠WARNING:

Water Heaters heat water to high temperatures. Temperatures over 125 Degrees F can cause severe burns instantly or death from scalds. While a HEWH operates in Consumer Mode, software applies measures that maintain the selected temperatures and protect the user from scald hazards or unintentional exposure to hot water. However, in Native Mode, software controls that are applied through algorithms are not active. Therefore it is the responsibility of the application programmer using Native Mode to follow all guidelines for Safe/Reliable Operation detailed below to protect the consumer from these scald hazards.

## Guidelines for Safe/Reliable Native Mode Operation

1. The user set point of the water heater is normally limited to 140 Degrees F. During **Native Mode** the user should not attempt to heat the water higher than 165 Degrees F as there is a mechanical Thermal Cut Out (TCO) that operates at approximately that temperature. TCO requires a manual reset if limits are exceeded.
2. Whenever a heating element is to be operated (Elements are: Compressor, Upper Electric Element, and Lower Electric Element) the Double Line Break (DLB) Relay should be closed/turned on. In order to maintain reliability of the DLB relay it is always closed while No heating elements are on and opened only after all heating elements are off. Control of the DLB is enforced during Native Mode when attempting to turn on heating elements. However, it may be overridden if you command the DLB directly.
3. Heating Element – Due to daisy chain wiring, one and only one heating element will be energized at any one time. When the first element is energized it deprives energy from the remaining elements. The priority within the daisy chain is Upper Electric Element, Lower Electric Element, and the Compressor.
4. Relay operation – Excessive relay cycling can shorten the life of the relays.
5. Mode Refresh – The Native mode must be asserted with a Native Mode (0x0A + 0x01, or 0xC0…) command at least one time every 5 minutes (2.5 minute periodic rate is recommended).

# Objective:

Documentation is intended to serve as basic instructions for a user interfacing to a Hybrid Electric Water Heater (HEWH). The maker module documentation related to the interfacing to a specific appliance will focus on the following areas:

1. Interaction Guidelines/Rules - Guidelines for using the maker module/green bean interface with HEWH products including:
   1. Consumer Mode Interactions – details how to use the product with the currently supported modes to add functionality.
   2. Native Mode Interactions – Details how to use the Native mode to design custom algorithms to add additional functionality to the unit.
2. Command Set – A detailed list of the commands that can be used to interface to the appliance in Consumer and Native modes.

# Interaction Guidelines:

Interaction between an external application and the Hybrid Electric Water Heater using the green bean, will take place in either a normal Consumer mode or in a Native/lower level mode. The mode choice depends upon the degree of control required. If a user want to extend the existing user interface to more easily control and monitor the behaviors that already exist, then Consumer mode interaction is sufficient. However, if different control algorithms are desired or operation beyond normal consumer controlled limits is required then Native mode of the control can be used.

## Consumer Mode Interaction:

In the consumer mode it is possible to modify the operational mode and the set temperature within normal operational limits. It is possible to query the current status to report back on unit progress etc. An external application can interact with the unit using the green bean to add automated features using existing operating modes. An application could extend the capabilities of the water heater when high load conditions are anticipated and maintain overall high energy efficiency as well. Consider the example of typical hot water demand where in the morning the water load for showering is high – but during the rest of the time the consumer want to operate the unit with energy performance in mind. The user could then operate the appliance according to a preprogrammed calendar to improve efficiency or hot water availability given the way they are using the water. An external application may choose to increase the water temperature so that the number of “hot” showers a unit can support is extended. Assume the following unit operation:

1. During normal operation:
   1. Water Heater is set to eHeat Mode
      1. Issue Set Mode command (0xEF, 0x14) for economy heat (0x03)
         1. [Source Address, 0xDF, 0x14, 0x03]
      2. Response will come back as
         1. [0x60, 0xDF, 0x14, 0x00]
   2. Set Point is 120 Degrees F.
      1. Issue a “Write Set Point” command (0xA5) with data being 120 (0x78)
         1. Command is [Source address, 0xA5, 0x78]
   3. This is the best performing mode for energy purposes but is not designed for maximum water availability or quickest recovery time.
2. The application Tracks the Time of Day and is preprogrammed to adjust the water temperature 3 hours before the normal morning routines begin (call that alarm time).
   1. At (Alarm Time – 3 hours)
      1. Application Adjusts the temperature to 140
         1. Issue a “Write Set Point” command (0xA5) with data being 140 (0x8C)
            1. Command is [Source address, 0xA5, 0x8C]
      2. Change the mode to Hybrid
         1. Issue Set Mode command (0xEF, 0x14) for Hybrid mode (0x00)
            1. [Source Address, 0xDF, 0x14, 0x00]
         2. Response will come back as
            1. [0x60, 0xDF, 0x14, 0x00]
   2. Changes allow for faster water heating time and extended shower times based on higher water temperatures and therefore less hot water usage for an individual shower.
3. Monitor the Tank temperatures as the alarm approaches and if the temperature is not responding as fast as desired it is possible to let the application choose a more aggressive heating strategy (Standard Electric). In addition the temperature profile can be stored so that the overall algorithm can be modified for better performance:
   1. Monitor the Tank Temperature every 5 minutes by issuing “Request Current Status” command (0xDE, 0x10). This can be used to confirm the heating mode as well as the tank temperature and set point.
   2. If we get to within 1 hour of the “Alarm Time” and the temperature is still more than 5 degrees from the target – switch to a more aggressive mode (Standard Electric) in order to promote heating.
      1. Issue Set Mode command (0xEF, 0x14) for Standard Electric Mode (0x01)
         1. [Source Address, 0xDF, 0x14, 0x01]
      2. Response will come back as
         1. [0x60, 0xDF, 0x14, 0x00]
4. The application intercedes again to adjust the operation back to normal modes when all of the showers have been completed and a slow recovery of water temperature is in order.
   1. @ (Alarm Time + 1 Hour)
      1. Adjust the set point to 120 Degrees F
         1. Issue a “Write Set Point” command (0xA5) with data being 120 (0x78)
            1. Command is [Source address, 0xA5, 0x78]
      2. Adjust the operating mode back to Economy Heat.
         1. Issue Set Mode command (0xEF, 0x14) for economy heat (0x03)
            1. [Source Address, 0xDF, 0x14, 0x03]
         2. Response will come back as
            1. [0x60, 0xDF, 0x14, 0x00]

The benefit or remaining in consumer mode is that the control is always monitoring the performance and reliable operation of the appliance while still giving the technical user a chance to customize performance to have maximum hot water when needed, while recovering with the lowest energy foot print.

## Native Mode Interaction:

In the Native mode the user/programmer can take a more active role in the control of the appliance. However, the reliability and the performance of the HEWH is the responsibility of the programmer and the algorithms they apply. The user is capable of individually controlling loads but as a result of this control they are responsible for following the **Guidelines for Safe/Reliable Native Mode Operation** (see above)**.**

Example use of Native Mode: Assume your goal is to heat water in the fastest possible way to a target temperature greater than 140 Degrees F (but less than 165) in order to increase the availability of water at a tap adjusted to 140. Control simulating standard electric modes for a higher target temperature using **Native Mode** would be as follows:

1. Enter Native Mode
   1. Issue Enter Native Mode command (See 0xC8)
2. Set Second Counter = 0
3. Start control loop – that runs ever second
   1. Increment Second Counter
   2. If Second Counter % SERVICE\_MODE\_REFRESH\_TIME (assume 2.5 minutes – 150 seconds)
      1. Refresh the Native Mode (service)
         1. Issue Native Mode Refresh Timeout (0xCB) command
   3. Read the Tank temperature
      1. Either Issue Request Current Status command (0xDE, 0x10)
      2. Or 🡪 Issue Request Hyper Terminal Data command (0x56)
   4. If the tank temperature >= desired temperature
      1. Turn all loads off
         1. Issue Native Mode Upper Element Control command 0xC9 – with the data being 0x00
            1. [Source address, 0xC9, 0x00]
         2. Native Mode Lower Element Control command (0xCA) with data being 0x00
            1. [Source address, 0xCA, 0x00]
   5. Else
      1. Calculate the temperature error (Desired temperature – Tank temperature)
      2. Keep track of loads that are on
         1. Either Read current system status or
            1. Issue Request Hyper Terminal Data command (0x56)
         2. Reference an application variables to track which loads are on
      3. If Upper element is currently On -- (see system status or application variable)
         1. if Error is <=5 degrees F
            1. Turn off the Upper heating element

Issue Native Mode Upper Element Control command 0xC9 – with the data being 0x00

[Source address, 0xC9, 0x00]

* + - * 1. Turn on the lower heating element

Issue Native Mode Upper Element Control command 0xC9 – with the data being 0xFF

[Source address, 0xCA, 0xFF]

* + - 1. Else
         1. Continue heating with upper element – may want to assert the load Turning on UE again:

Turn on the Upper heating element

Issue Native Mode Upper Element Control command 0xC9 – with the data being 0xFF

[Source address, 0xC9, 0xFF]

* + 1. Else – if error is > 10 degrees
       1. Turn off lower element
          1. Native Mode Lower Element Control command (0xCA) with data being 0x00

[Source address, 0xCA, 0x00]

* + - 1. Turn on Upper control element
         1. Issue Native Mode Upper Element Control command 0xC9 – with the data being 0xFF

[Source address, 0xC9, 0xFF]

* + 1. Else - if error > 5 degrees && no element is currently on.
       1. Turn on the Lower Element
          1. Native Mode Lower Element Control command (0xCA) with data being 0xFF
          2. [Source address, 0xCA, 0xFF]
  1. Use application to for 1 second
  2. Repeat Loop

# Command Set:

Each command that the oven supports will be discussed in general terms. For each command it is important that the user understand the destination address, the command itself (whether it is single or multi-byte), and the additional data that is sent with the command.

Destination Address when dealing with for the HEWH is 0x60.

When a command is transmitted it is sent with a source address. Source address may not matter and will only be discussed if the command logic takes the source address into account - “Data” area of the command below will generally be talking about what is added after the basic command.

The command set is as follows:

1. Read Software Version –
   1. Purpose: Query response to the control to determine current version of the application software.
   2. Command: 0x01
   3. Data sent with command (beyond source address and command): NA
   4. Response: Current Software major and minor versions. The software version is maintained as AA.BB.CC.DD where each AA-DD are just ## values appears within the range of 00 – 99. The data within the response is represented as a binary/hex value e.g. 99 = 0x63.
   5. Example Query
      1. [Source address, 0x01]
   6. Example Response: for version 04.02.03.99 –
      1. [0x60, 0x01, 0x01,0x02,0x03,0x63] where:
         1. 0x60 = Source address of control
         2. 0x01 = command (version response – distinguished from query using total length of packet)
         3. 0x04 = Version AA = 04
         4. 0x02 = Version BB = 02
         5. 0x03 = Version CC = 03
         6. 0x63 = Version DD = 99
2. Get Tank Thermistor Resistance Value (T2)
   1. Purpose: request the equivalent resistance value based on the present ADC reading for a given thermistor. Value returned will vary between 0 and 99999 ohms.
   2. Command: 0x10
   3. Data sent with command (beyond source address and command): NA
   4. Response: Resistance of thermistor requested represented as a 32-bit quantity.
   5. Example Command:
      1. [Source address, 0x10]
   6. Example Response: Assume resistance is 74000 ohms – 0x00 01 21 10
      1. [0x60, 0x10, 0x00, 0x01, 0x21, 0x10]
3. Get Evaporator Input Thermistor Resistance Value (T3A)
   1. Purpose: request the equivalent resistance value based on the present ADC reading for a given thermistor.
   2. Command: 0x11
   3. Data sent with command (beyond source address and command): NA
   4. Response: Resistance of thermistor requested represented as a 32-bit quantity.
   5. Example Command:
      1. [Source address, 0x11]
   6. Example Response: Assume resistance is 74000 ohms – 0x00 01 21 10
      1. [0x60, 0x11, 0x00, 0x01, 0x21, 0x10]
4. Get Evaporator Output Thermistor Resistance Value (T3B)
   1. Purpose: request the equivalent resistance value based on the present ADC reading for a given thermistor.
   2. Command: 0x12
   3. Data sent with command (beyond source address and command): NA
   4. Response: Resistance of thermistor requested represented as a 32-bit quantity.
   5. Example Command:
      1. [Source address, 0x12]
   6. Example Response: Assume resistance is 74000 ohms – 0x00 01 21 10
      1. [0x60, 0x12, 0x00, 0x01, 0x21, 0x10]
5. Get Compressor Thermistor Resistance Value (TMXTRA/T4)
   1. Purpose: request the equivalent resistance value based on the present ADC reading for a given thermistor.
   2. Command: 0x13
   3. Data sent with command (beyond source address and command): NA
   4. Response: Resistance of thermistor requested represented as a 32-bit quantity.
   5. Example Command:
      1. [Source address, 0x13]
   6. Example Response: Assume resistance is 74000 ohms – 0x00 01 21 10
      1. [0x60, 0x13, 0x00, 0x01, 0x21, 0x10]
6. Get Ambient Thermistor Resistance Value (T5)
   1. Purpose: request the equivalent resistance value based on the present ADC reading for a given thermistor.
   2. Command: 0x14
   3. Data sent with command (beyond source address and command): NA
   4. Response: Resistance of thermistor requested represented as a 32-bit quantity.
   5. Example Command:
      1. [Source address, 0x14]
   6. Example Response: Assume resistance is 74000 ohms – 0x00 01 21 10
      1. [0x60, 0x14, 0x00, 0x01, 0x21, 0x10]
7. Get Shunt Current Resistance (IDLB)
   1. Purpose: request the equivalent resistance value based on the present ADC reading for a given thermistor.
   2. Command: 0x15
   3. Data sent with command (beyond source address and command): NA
   4. Response: Resistance of thermistor requested represented as a 32-bit quantity.
   5. Example Command:
      1. [Source address, 0x15]
   6. Example Response: Assume resistance is 74000 ohms – 0x00 01 21 10
      1. [0x60, 0x15, 0x00, 0x01, 0x21, 0x10]
8. Get Personality Model Select
   1. Purpose: Request the ADC value associated with the model selection. Value will be a 16-bit quantity representing the average ADC reading.
   2. Command: 0x16
   3. Data sent with command (beyond source address and command): NA
   4. Response: Average ADC value returned as a 32-bit quantity –actually only 16 bits are populated here.
   5. Example Command:
      1. [Source address, 0x16]
   6. Example Response: Assume ADC value is 1000 (0x03E8)
      1. [0x60, 0x16, 0x00, 0x00, 0x03, 0xE8]
9. Read RAM Memory @ Address (model dependent)
   1. Purpose: Read data located at a given RAM address – normally used for debug only.
   2. Command: 0x4E
   3. Data sent with command (beyond source address and command): 5 bytes as follows:
      1. Byte 1 = number of bytes to read up to 1-16
      2. Bytes 2-5 = 4 byte address – MSB – LSB -- address must be in RAM space from 0x0400 <= x <= 0x21FF
   4. Response: Response includes a fault error code followed by the address and the data requested. Decoding as follows:
      1. Error code 0 = All pass, 0xFF = invalid address, 0xFE = invalid number of bytes requested)
      2. 32-bit address... MSB to LSB
      3. N - Data bytes requested.
   5. Example Command: Read 8 bytes of data at address 0x0400
      1. [Source address, 0x4E,0x08,0x00, 0x00, 0x04,0x00]
   6. Example Response: Data located at address 0x1000 - -assume 0x01 – 0x08 stored here…
      1. [0x60, 0x4E, 0x00, 0x00, 0x00, 0x04, 0x00, 0x01, 0x02, 0x03, 0x04, 0x05 0x06, 0x07, 0x08]
10. Read Data Flash Record (model dependent)
    1. Purpose: Request data stored in 1 of 8 different data flash records. Format of each of the 8 records varies. Used for debug and confirmation of stored data. Records are used for the following:
       1. 1 = user modified data including set point, mode and delay settings…
       2. 2 = Error counters related to operating condition changes…
       3. 3 = control limits for unit including minimum and maximum temperature set points, vacation days etc…
       4. 4 = parametric data controlling some diagnostics
       5. 5 = Mode control parameters.
       6. 6 = Heating zone and flow event parameters…
       7. 7 = Model and Serial Number storage
       8. 8 = Models supported information
    2. Command: 0x4F
    3. Data sent with command (beyond source address and command): Data flash record, offset, and the number of bytes from that record that are to be read.
       1. 1 –byte coded as 1 – 8 indicating which data flash record to access
       2. O – offset within the record to return the data 0 - 255
       3. N – Number of bytes within that record that is being requested (1 – 16)
    4. Response: Response includes a fault error code followed by the record number, offset, and the N-data bytes located at that offset. Decoding as follows:
       1. Error code 0 = All pass, 0xFF = invalid record number, 0xFE = invalid number of bytes requested)
       2. Record Number…
       3. O – offset within the record
       4. N - Data bytes requested.
    5. Example Command: Read 8 bytes of data from record 3 at offset 16.
       1. [Source address, 0x4F, 0x03, 0x10, 0x08]
    6. Example Response: 8 bytes of data from record 3 at offset 16 … (assume 0x01 – 0x08 are stored…)
       1. [0x60, 0x4F, 0x00, 0x03, 0x01, 0x02, 0x03, 0x04, 0x05 0x06, 0x07, 0x08]
11. Read Sealed System Fault Counters
    1. Purpose: Request to read current count values of any sealed system operational faults that have occurred. Faults represent conditions when sealed system behavior was altered due to sensed data – not necessarily that the sealed system has any hardware issues.
    2. Command: 0x53
    3. Data sent with command (beyond source address and command): NA
    4. Response: Response includes a fault error counters for 9 different operational conditions (A – I).
    5. Example Command:
       1. [Source address, 0x53]
    6. Example Response: Assume counts vary from 1 - 10
       1. [0x60, 0x53, 0x01, 0x02, 0x03, 0x04, 0x05 0x06, 0x07, 0x08,0x09, 0x0A]
12. Read Miscellaneous Fault Counters
    1. Purpose: Request to read current count values of miscellaneous faults.
    2. Command: 0x54
    3. Data sent with command (beyond source address and command): NA
    4. Response: Response includes a fault error counters maintained for 14 items - fan, compressor, upper element, lower element, dirty filter, 5-thermistors, record write, stuck key, dry tank and miss wire failures.
    5. Example Command:
       1. [Source address, 0x54]
    6. Example Response: Assume counts vary from 1 - 14
       1. [0x60, 0x54, 0x01, 0x02, 0x03, 0x04, 0x05 0x06, 0x07, 0x08,0x09, 0x0A,0x0B, 0x0C, 0x0D, 0x0E]
13. Clear Fault Counters
    1. Purpose: Command sent to clear all fault counters. Generally sent as part of a service activity to track effectiveness of repair.
    2. Command: 0x55
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source address, 0x55]
    6. Example Response: NA
14. Request Hyper Terminal Data
    1. Purpose: Request debug output stream. Representing a snapshot of current unit operation.
    2. Command: 0x56
    3. Data sent with command (beyond source address and command): NA
    4. Response: 37 bytes of data + address and command. Data returned includes
       1. Upper Element Relay status (1/0 = On/Off)
       2. Lower Element Relay status (1/0 = On/Off)
       3. Compressor Relay status (1/0 = On/Off)
       4. Fan Relay status (1/0 = On/Off)
       5. 8 – bit - User Set point
       6. 16 bit - Tank Temperature
       7. 16 – bit – T3A temperature
       8. 16 – bit – T3B temperature
       9. 16 – bit – T4 temperature
       10. 16 – bit – T5 temperature
       11. DLB current
       12. 16-bit AC voltage
       13. 16- bit - EEV valve position
       14. 8-bit coded flow even stats (lowest three bits indicating small, medium and large flow events active
       15. 8- bit current operational mode
       16. 16-bit – code for termination of last heating cycle.
       17. 8-bit - next heating element to control
       18. 8-bit - compressor minimum operational time state
       19. 8-bit - compressor FSM state 9related to internal implementation)
       20. 8- bit - valve FSM state (internal implementation)
    5. Example Command:
       1. [Source address, 0x56]
    6. Example Response: no detail provided
       1. [0x60, 0x56, 37 bytes….]
15. Request Thermistor Resistance Values
    1. Purpose: Command sent request all of the resistance values in one command.
    2. Command: 0x80
    3. Data sent with command (beyond source address and command): NA
    4. Response: Data returned is 7 different 16-bit quantities that represent the resistance values for IDLB (current sense), T2 (tank), T5 (ambient), T4 (compressor), T3A (evap input), T3b (evap output), Model Selection.
    5. Example Command:
       1. [Source address, 0x80]
    6. Example Response: No details see earlier commands for details on the return values. Note that each is scaled to 16-bits by dividing the max by 10?
16. Reset Filter Times
    1. Purpose: Command will reset the filter time counter.
    2. Command: 0x90
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source address, 0x90]
    6. Example Response: NA
17. Write Set Point
    1. Purpose: Adjust the current temperature set point. May be used for limited remote control.
    2. Command: 0xA5
    3. Data sent with command (beyond source address and command): 1 byte representing the set point in Degrees F. The change is declined if the water heater is in a Demand Response mode. Set point adjustment is bounded by a parametric minimum and maximum value. Normally this is between 60 and 140 Degrees F. Normally the display of the Water heater will transition to a mode where the set point is displayed.
    4. Response: NA
    5. Example Command: Setting the Water heater to 120 (0x78) Degrees F.
       1. [Source address, 0xA5, 0x78]
    6. Example Response: NA
18. Double Line Break (DLB) Relay Commands (Toggle/On/Off)
    1. Purpose: Control DLB relay for test purposes. Used only in Native modes. Command will force control into Native mode if not currently in that mode. Command will either Toggle, Close, or Open DLB Relay. If the DLB is off then all other heating elements are also turned off before DLB turned on. If the DLB is turned on, it will remain on only for 5 minutes before automatically turning off DLB.
    2. Commands: Toggle (0xB0) / Turn On (0xB1)/ Turn Off (0xB2)
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command: Toggling DLB relay state
       1. [Source address, 0xB0]
    6. Example Response: NA
19. Enter/Exit Native Mode (FCT)
    1. Purpose: Forces control to enter/exit Native mode. Mode is used for individual relay control. On Native mode entry all relays will turn off, the Display will show “888”, and the buzzer will sound. When Native mode is exited the control will reset.
    2. Command: 0xB4
    3. Data sent with command (beyond source address and command): 1 byte indicating whether to Enter (0xFF) or Exit (any other byte) Native mode.
    4. Response: NA
    5. Example Command: Entering Native Mode.
       1. [Source address, 0xB4, 0xFF]
    6. Example Response: NA
20. SET LEDs (model dependent)
    1. Purpose: Controls the display values in Native mode.
    2. Command: 0xB5
    3. Data sent with command (beyond source address and command): 5 bytes indicating the status of the LEDs that are part of the display. The first 3 bytes relate to the Seven Segment display characters (left, center, right). The remaining 2 bytes refer to the discrete LEDs outside of the seven segment characters.
       1. Seven Segment Display Characters are mapped as follows: also an fyi see 🡪 <http://www.learningaboutelectronics.com/Articles/How-to-display-any-character-on-a-7-segment-LED-display.php> )

a a a a a a

f b

f b

ggggggggg

e c

e c

dddddddd

* + - 1. Where a – g are mapped from bit 1 –bit 7 of the individual byte. To display a ‘4’ for instance the byte code would be 0xCC (segments b (0x04),c (0x08) ,f (0x40) ,g (0x80) on)
    1. Remaining LEDs are individually mapped as follows:
    2. Byte 4/5 – bits are mapped:
       1. Bit 1 = Z16 and Z2 (Mode LEDs)
       2. Bit2 = Z3 (Degrees F)
       3. Bit3 = Z4 (Days\_2)
       4. Bit4 = Z5 (Days\_1)
       5. Bit5 = Z6 (ENTER)
       6. Bit6 = Z7 (HYBRID)
       7. Bit7 = Z8 (EHEAT)
    3. Byte 5/5 – bits are mapped:
       1. Bit 1 = Z9 (VACATION)
       2. Bit2 = Z10 (STD\_ELECTRIC)
       3. Bit3 = Z11 (HIGH\_DEMAND)
       4. Bit4 = Z12 (MID\_LOW)
       5. Bit5 = Z13 (FILTER\_HIGH)
       6. Bit6 = Z14 (FILTER\_LOW)
       7. Bit7 = Z15 (DEGREES\_C)
  1. Response: NA
  2. Example Command: Displaying ‘1’ (segment b,e – 0x0C), ‘2’ (segments a,b,d,e,f – 0xE6, ’3’ (segments a,b,c,d,g – 0xAE on the Seven Segment display and turning on only the Degrees F (0x04) and Vacation LEDs (0x02)
     1. [Source address, 0xB5, 0x0C, 0xE6, 0xAE, 0x04, 0x02]
  3. Example Response: NA

1. Set Electronic Expansion Valve Control Lines – (model specific)
   1. Purpose: Commands to drive the stepper motor for the Electronic Expansion Valve. Command should only be issued in Native mode. Normally used for test purposes. At other times the EEV is controlled by a PID algorithm for optimum performance.
   2. Command: 0xB6
   3. Data sent with command (beyond source address and command): 1 byte containing the status of each of the 4 control lines. (Note remapped internally A, B, A\_Bar, B\_Bar) : Bit encoded as follows:
      1. Line\_A – bit 0
      2. Line\_A\_Bar – bit 1
      3. Line B – bit 2
      4. Line\_B\_Bar – bit 3
   4. Response: NA
   5. Example Command: Setting both A and B High while holding A\_Bar and B\_Bar low.
      1. [Source address, 0xB6, 0x05]
   6. Example Response: NA
2. Read Key Status
   1. Purpose: Read the current Press Release status of all of the keys
   2. Command: 0xB7
   3. Data sent with command (beyond source address and command): NA
   4. Response: 2 bytes of data that are bit mapped to the current key status as follows:
      1. MSB
         1. Bit 0 = Switch 1 (Up Key)
         2. Bit 1 = Switch 2 (Enter Key)
         3. Bit 2 = Switch 3 (Down Key)
         4. Bit 3 = Switch 4 (Unmapped Key)
         5. Bit 4 = Switch 5 (Mid High Key)
         6. Bit 5 = Switch 6 (Filter Key)
         7. Bit 6 = Switch 7 (Alternate Mode Key)
         8. Bit 7 = Switch 8) (Unmapped Key)
      2. LSB
         1. Bit 0 = Switch 9 (Alternate Filter Key)
         2. Bit 1 = Switch 10 (Mode Key)
         3. Bit 2 = Switch 11 (Alternate Mid Low Key)
         4. Bit 3 = Switch 12 (Unmapped Key)
         5. Bit 4 = Switch 13 (Unmapped Key)
         6. Bit 5 = Switch 14 (Unmapped Key)
         7. Bit 6 = Switch 15 (Unmapped Key)
         8. Bit 7 = Switch 16 (Unmapped Key)
   5. Example Command:
      1. [Source address, 0xB7]
   6. Example Response: Indicating that the Enter Key is pressed and all other keys are released.
      1. [0x60, 0xB7, 0x02, 0x00]
3. Read Personality
   1. Purpose: Returns model type/personality of the unit.
   2. Command: 0xBC
   3. Data sent with command (beyond source address and command): NA
   4. Response: 1 byte of data indicating the model found or 0xFF if model is invalid or yet to be determined. Normally the model number will be from 0 – 15.
   5. Example Command:
      1. [Source address, 0xBC]
   6. Example Response: Indicating that it is model 2.
      1. [Source address, 0xBC, 0x02]
4. Enter Native Mode (Factory use)
   1. Purpose: Command to enter Native mode. On entry all heat control loads are turned off. When Native mode is entered the Seven Segment display will show “888”.
   2. Command: 0xC0
   3. Data sent with command (beyond source address and command): NA
   4. Response: NA
   5. Example Command:
      1. [Source address, 0xC0]
   6. Example Response: NA
5. Exit Native Mode (Factory Use)
   1. Purpose: Exit Native test mode. When the command is received the control is reset.
   2. Command: 0xC1
   3. Data sent with command (beyond source address and command): NA
   4. Response: NA
   5. Example Command:
      1. [Source address, 0xC1]
   6. Example Response: NA
6. Restore EEPROM Defaults
   1. Purpose: Sets all values to factory defaults. Including resetting error counts, setting the heating mode to hybrid, and the control temperature to 120 Degrees F. Used as part of production process.
   2. Command: 0xC2
   3. Data sent with command (beyond source address and command): NA
   4. Response: NA
   5. Example Command:
      1. [Source address, 0xC2]
   6. Example Response: NA
7. Set Model Number
   1. Purpose: Used to set the model stored in the Virtual EEPROM. Model is used for identification and software update only. Model set here does not affect the performance of the unit.
   2. Command: 0xC3
   3. Data sent with command (beyond source address and command): Up to 33 characters that would be equivalent to the model code on the model plate.
   4. Response: NA
   5. Example Command:
      1. [Source address, 0xC3, up to 33 characters…]
   6. Example Response: NA
8. Set Serial Number
   1. Purpose: Used to set the serial number stored in the Virtual EEPROM. Used for identification and potentially software update only. Serial Number set here does not affect the performance of the unit.
   2. Command: 0xC4
   3. Data sent with command (beyond source address and command): Up to 33 characters that would be equivalent to the model code on the model plate.
   4. Response: NA
   5. Example Command:
      1. [Source address, 0xC4, up to 33 characters…]
   6. Example Response: NA
9. Request Model Number
   1. Purpose: Used to read the model stored in the Virtual EEPROM. Model is used for identification and software update only.
   2. Command: 0xC5
   3. Data sent with command (beyond source address and command): NA
   4. Response: Up to 33 characters that would be equivalent to the model code on the model plate.
   5. Example Command:
      1. [Source address, 0xC5]
   6. Example Response: [0x60, 0xC5, 33 characters stored in Virtual EEPROM]
10. Get Serial Number
    1. Purpose: Used to read the serial number stored in the Virtual EEPROM.
    2. Command: 0xC6
    3. Data sent with command (beyond source address and command): NA
    4. Response: Up to 33 characters that would be equivalent to the serial number on the model plate.
    5. Example Command:
       1. [Source address, 0xC6]
    6. Example Response: [0x60, 0xC6,, 33 characters stored in Virtual EEPROM]
11. Enter Native (Service)
    1. Purpose: Enter/Exit Native (Service) mode. On entry a tone is sounded, heating loads are turned off and a 15 minute timer is started. If the timer expires without a Native mode interaction then control will revert to consumer mode.
    2. Command: 0xC8
    3. Data sent with command (beyond source address and command): 1 byte - 0xFF indicating to enter the mode – any other value is to exit Native mode.
    4. Response: NA
    5. Example Command:
       1. [Source address, 0xC8, 0xFF]
    6. Example Response: NA
12. Native Mode Upper Element Control (model dependent)
    1. Purpose: Turn on or off the Upper element while in Native mode. The element is only turned on for a limited period of time (approximately 10 minutes) before it will automatically be turned off.
    2. Command: 0xC9
    3. Data sent with command (beyond source address and command): 1 byte - ON/OFF (0xFF/0x00).
    4. Response: NA
    5. Example Command: Turning on the upper element.
       1. [Source address, 0xC9, 0xFF]
    6. Example Response: NA
13. Native Mode Lower Element Control (model dependent)
    1. Purpose: Turn on or off the Lower element while in Native mode. The element is only turned on for a limited period of time (approximately 10 minutes) before it will automatically be turned off.
    2. Command: 0xCA
    3. Data sent with command (beyond source address and command): 1 byte - ON/OFF (0xFF/0x00).
    4. Response: NA
    5. Example Command: Turning on the element.
       1. [Source address, 0xCA, 0xFF]
    6. Example Response: NA
14. Native Mode Refresh Timeouts (Service)
    1. Purpose: Resets timeouts associated with load control and retention of the Native mode (service). Used to maintain element drive or time in mode.
    2. Command: 0xCB
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source address, 0xCB]
    6. Example Response: NA
15. Request System Mode
    1. Purpose: Query to confirm the current mode of operation.
    2. Command: 0xCE
    3. Data sent with command (beyond source address and command): NA
    4. Response: Single byte indicating the mode as follows:
       1. 0 = Consumer Mode (default mode after reset)
       2. 1 = Native (Factory) Mode
       3. 2 = Native (Service) Mode
       4. 3 = Native (FCT) Mode
       5. 4 = Demo Mode (similar to consumer mode but no loads would be activated)
    5. Example Command:
       1. [Source address, 0xCE]
    6. Example Response: Example shows unit in consumer mode
       1. [0x60, 0xCE, 0x00]
16. Set EEV PID Proportional Denominator
    1. Purpose: Used to adjust the proportional gain during tuning. Value should always be non-zero and if it is changed the value is stored in non-volatile FLASH and will alter the performance of the electronic expansion valve.
    2. Command: 0xE0
    3. Data sent with command (beyond source address and command): 16-bit Value representing the denominator to use.
    4. Response: NA
    5. Example Command: Setting Denominator to 100 – 0x0064
       1. [Source address, 0xE0, 0x00]
    6. Example Response: NA
17. Set EEV PID Integral Denominator @ Low Ambient
    1. Purpose: Used to adjust the integration gain during tuning. Value is used at low ambient temperature conditions. Value should always be non-zero and if it is changed the value is stored in non-volatile FLASH and will alter the performance of the electronic expansion valve.
    2. Command: 0xE1
    3. Data sent with command (beyond source address and command): 16-bit Value representing the denominator to use.
    4. Response: NA
    5. Example Command: Setting Denominator to 100 – 0x0064
       1. [Source address, 0xE1, 0x00, 0x64]
    6. Example Response: NA
18. Set EEV PID Integral Denominator @ High Ambient
    1. Purpose: Used to adjust the integration gain during tuning. Value is used at high ambient temperature conditions. Value should always be non-zero and if it is changed the value is stored in non-volatile FLASH and will alter the performance of the electronic expansion valve.
    2. Command: 0xE2
    3. Data sent with command (beyond source address and command): 16-bit Value representing the denominator to use.
    4. Response: NA
    5. Example Command: Setting Denominator to 100 – 0x0064
       1. [Source address, 0xE1, 0x00, 0x64]
    6. Example Response: NA
19. Set EEV Start Position @ Low Ambient
    1. Purpose: Used to declare the valve starting/home position during low ambient conditions. Value will be something approaching 480. If it is changed the value is stored in non-volatile FLASH and will alter the performance of the electronic expansion valve. Must be <= 480.
    2. Command: 0xE3
    3. Data sent with command (beyond source address and command): 16-bit Value
    4. Response: NA
    5. Example Command: Setting to 480 – 0x01E0
       1. [Source address, 0xE3, 0x00, 0x01E0]
    6. Example Response: NA
20. Set EEV Start Position @ High Ambient
    1. Purpose: Used to declare the valve starting/home position during high ambient conditions. Value will be something approaching 480. If it is changed the value is stored in non-volatile FLASH and will alter the performance of the electronic expansion valve.
    2. Command: 0xE4
    3. Data sent with command (beyond source address and command): 16-bit Value
    4. Response: NA
    5. Example Command: Setting to 480 – 0x01E0
       1. [Source address, 0xE4, 0x01, 0x0E0]
    6. Example Response: NA
21. Set EEV Stop Position
    1. Purpose: Used to declare the valve stop position. EEV will not be adjusted beyond that count. Must be less than or equal to 480.
    2. Command: 0xE5
    3. Data sent with command (beyond source address and command): 16-bit Value representing the denominator to use.
    4. Response: NA
    5. Example Command: Setting to 256
       1. [Source address, 0xE4, 0x01, 0x00]
    6. Example Response: NA
22. Set Ambient Superheat Threshold1 - 3
    1. Purpose: 1 of 3 different ambient operating temperatures used to adjust the SuperHeat temperature that is targeted by control of the EEV.
    2. Command: 0xE6, 0xE7, 0xE8 (for threshold 1,2,3 respectively)
    3. Data sent with command (beyond source address and command): 8-bit Value in Degrees F.
    4. Response: NA
    5. Example Command: Setting to threshold 1 to 64 - 0x40
       1. [Source address, 0xE6, 0x40]
    6. Example Response: NA
23. Set Low Super Heat Target
    1. Purpose: Setting super heat temperature that the ambient temperature is below the low superheat ambient threshold. Superheat targets when the ambient temperature falls in other regions defined by the threshold are interpolated between the Low and maximum superheat targets. Command is used for performance tuning. Superheat temperatures are maintained through control of the EEV.
    2. Command: 0xE9
    3. Data sent with command (beyond source address and command): 8-bit Value in Degrees F.
    4. Response: NA
    5. Example Command: Setting to 64 - 0x40
       1. [Source address, 0xE9, 0x40]
    6. Example Response: NA
24. Set Mid Super Heat Target
    1. Purpose: Setting super heat temperature that the ambient temperature is greater than the low superheat ambient threshold. Superheat targets are interpolated between the Low and mid superheat targets. Command is used for performance tuning. Superheat temperatures are maintained through control of the EEV.
    2. Command: 0xEA
    3. Data sent with command (beyond source address and command): 8-bit Value in Degrees F.
    4. Response: NA
    5. Example Command: Setting to 64 - 0x40
       1. [Source address, 0xEA, 0x40]
    6. Example Response: NA
25. Get Super Heat Target
    1. Purpose: Query the super heat target that is currently applied. Superheat temperatures are maintained through control of the EEV.
    2. Command: 0xEC
    3. Data sent with command (beyond source address and command): NA
    4. Response: 16-bit temperature Deg F \* 10. E.g. if target is 100, 1000 is returned.
    5. Example Command:
       1. [Source address, 0xEC]
    6. Example Response: Assume target returned is 1000
       1. [0x60, 0xEC, 0x03, 0xE8]
26. Get PID Parameters
    1. Purpose: Query the current parameters that are applied to the PID control of the EEV.
    2. Command: 0xEF
    3. Data sent with command (beyond source address and command): NA
    4. Response: contains all of the PID parameters as follows:
       1. 16-bit Proportional Denominator
       2. 16-bit low ambient Integral term denominator
       3. 16-bit high ambient Integral term denominator
       4. 16-bit low ambient EEV start position
       5. 16-bit high ambient EEV start position
       6. 16-bit EEV stop position
       7. 8-bit First EEV superheat threshold Temperature Degrees F
       8. 8-bit Second EEV superheat threshold Temperature Degrees F
       9. 8-bit Third EEV superheat threshold Temperature Degrees F
       10. 8-bit Low Superheat Temperature Degrees F
       11. 8-bit Mid Superheat Temperature Degrees F
    5. Example Command:
       1. [Source address, 0xEF]
    6. Example Response: Proportional denominator set to 100…
       1. [0x60, 0xEF, 0x00, 0x64…]
27. Native Mode – Compressor Off
    1. Purpose: Turn compressor off during Native modes (factory or service).
    2. Command: 0x30
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source address, 0x30]
    6. Example Response: NA
28. Native Mode – Fan Control
    1. Purpose: Turn Fan on/off (0xFF/0x00) off during Native Mode (factory, service, or FCT). If fan is turned on in a Native mode – it will be automatically turned off in 10 minutes unless the timer has been reset (by sending command again or refreshing timers).
    2. Command: 0x31
    3. Data sent with command (beyond source address and command): On/Off (0xFF/0x00)
    4. Response: NA
    5. Example Command: Turning fan on
       1. [Source address, 0x31, 0xFF]
    6. Example Response: NA
29. Request LED Status
    1. Purpose: Returns the current On/Off status of all of the LEDs on the UI.
    2. Command: 0X32
    3. Data sent with command (beyond source address and command): NA
    4. Response: LED bytes are coded within 5 bytes of data. Same encoding represented in command SET LEDs (0xB5)
    5. Example Command:
       1. [Source address, 0x32]
    6. Example Response: see example in Set LEDs for encoding – Currently displaying ‘1’ (segment b,e – 0x0C), ‘2’ (segments a,b,d,e,f – 0xE6, ’3’ (segments a,b,c,d,g – 0xAE on the Seven Segment display and turning on only the Degrees F (0x04) and Vacation LEDs (0x02)
       1. [0x60, 0xF2, 0x0C, 0xE6, 0xAE, 0x04, 0x02]
30. Upper/Lower Element On Time Terminated
    1. Purpose: Turns on the upper/Lower element in the system for a brief time interval. Time interval is 1 second + value stored as parametric data as an 8-bit value in tenths of seconds from 0 – 25.5 seconds. Allowed only in Native mode.
    2. Command: 0x33/0x34
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command: Command for upper element on…
       1. [Source address, 0x33]
    6. Example Response: NA
31. Start Factory Compressor Test
    1. Purpose: Native mode (factory) command that initiates a compressor test sequence that holds the compressor on for 15 minutes. Used to test pull down operation.
    2. Command: 0x35
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source address, 0x35]
    6. Example Response: NA
32. Request Temperature Range
    1. Purpose: Used to obtain the minimum and maximum set points for a given unit.
    2. Command: 0XF7
    3. Data sent with command (beyond source address and command): NA
    4. Response: 2 bytes indicating minimum and max set points in degrees F
    5. Example Command:
       1. [Source address, 0xF7]
    6. Example Response: (min at 50 and max at 140)
       1. [0x60, 0xF7, 0x32, 0x8C]
33. Request Vacation Range
    1. Purpose: Used to obtain the minimum and maximum days for vacation mode.
    2. Command: 0xF8
    3. Data sent with command (beyond source address and command): NA
    4. Response: 2 bytes indicating minimum and max days
    5. Example Command:
       1. [Source address, 0xF7]
    6. Example Response: (min at 1 and max at 14)
       1. [0x60, 0xF7, 0x01, 0x0E]
34. Request Water Capacity
    1. Purpose: Used to obtain water capacity for unit.
    2. Command: 0xFA
    3. Data sent with command (beyond source address and command): NA
    4. Response: 1 byte indicating tank size in gallons
    5. Example Command:
       1. [Source address, 0xF7]
    6. Example Response: (50 gallon)
       1. [0x60, 0xF7, 0x3C]
35. Cancel EEV Position
    1. Purpose: Sets Valve position to invalid. End result of this is to force the valve to go back to the start position.
    2. Command: 0xFB
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source address, 0xFB]
    6. Example Response: NA
36. Command EEV Position
    1. Purpose: Commands to move to desired step count position.
    2. Command: 0xFC
    3. Data sent with command (beyond source address and command): 16-bit valve position. Must be <= 480 to avoid going back to the start position
    4. Response: NA
    5. Example Command: setting to 480
       1. [Source address, 0xFC, 0x01, 0xE0]
    6. Example Response: NA
37. Request Supported Modes
    1. Purpose: What modes are supported by unit?
    2. Command: 0xFE
    3. Data sent with command (beyond source address and command): NA
    4. Response: 1-byte that has bit encoding indicating the modes of operation that the unit supports. Bits encoded as follows:
       1. Bit 4 = Vacation Mode
       2. Bit 3 = High demand Mode
       3. Bit 2 = Economy Heat Mode (e-heat – compressor only)
       4. Bit 1 = Standard Electric
       5. Bit 0 = Hybrid Mode
    5. Example Command:
       1. [Source address, 0xFE]
    6. Example Response: All modes except vacation supported.
       1. [0x60, 0xFE, 0x0F]
38. Read Boot Loader Parametric Data
    1. Purpose: Read one of the allocated write once read many locations maintained by the boot loader.
    2. Command: 0xDD 0x03
    3. Data sent with command (beyond source address and command): Parametric Index of value requested. The parametric data corresponding to the parametric index varies by model but the following indexes are fixed:
       1. 0 = nature of the boot loader parametric data
       2. 1 = Board SKU
       3. 2 = Board Serial Number
       4. 3 = Unit SKU (Model Number)
       5. 4 = Unit Serial Number
    4. Response: Data that represents the parameter or an indication that the data offset requested was invalid. Note that when requesting 0 it is a special case where the return value is an indication of the number of parametric items that exist and the size.
       1. If 0 is requested…then data after command is
          1. N – Byte = number of items that are maintained
          2. S – byte – size of the items (same for all items)
       2. If non-zero valid offset is requested
          1. X – byte – index of parametric item requested (1 – N)
          2. S bytes – data stored at parametric item
    5. Example Command: Example is a request for the Model number ERD.
       1. [Source Address, 0xDD,0x03,0x01]
    6. Example Response:
       1. [0x60, 0xDD, 0x03 0x01, Data…]
39. Jump to Boot Loader
    1. Purpose: Forces Application to jump to the Boot loader application for possible software update. Command will only be acted upon if the unit is in a consumer standby (no cooking) or test mode.
    2. Command: 0xDD, 0x07
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [0x60, 0xDD,0x07]
    6. Example Response: NA
40. Software Image Valid
    1. Purpose: Allows user to query the validity of one of the software images stored within the control.
    2. Command: 0xDD 0x08
    3. Data sent with command (beyond source address and command): Image ID for the header requested where:
       1. 1 = Boot loader
       2. 2 = Main Application
       3. 3 = Parametric Data
       4. 4 = Auxiliary Image
    4. Response: A structure representing whether the image id was in range and the header date or just that the id was out of range.
       1. Either Case includes:
          1. Byte -- Image ID
          2. Byte – Valid Range (1/0)
          3. U16 – 0x0000 (intended for CRC of header but currently always 0x0000)
       2. If invalid (Valid Range = 1) the remainder of the data is the header for the image – but no more than the max packet that can be transmitted.
    5. Example Command: Example is a request for the Main Application header
       1. [Source Address, 0xDD, 0x08, 0x02]
    6. Example Response:
       1. [0x60, 0xDD, 0x08, 0x03,0x01, 0x00,0x00,header data…]
41. Stop Debug Messages/Heart Beat
    1. Purpose: Forces Application to stop publishing status information without being queried. Unit will stop doing so for 5 minutes.
    2. Command: 0xDD, 0x0D
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source Address, 0xDD,0x08]
    6. Example Response: NA
42. Request Demand Response Override Status
    1. Purpose: Query to see if the user has overridden any demand response modes.
    2. Command: 0xDE, 0x02
    3. Data sent with command (beyond source address and command): NA
    4. Response: Single byte indicating the override status(Full/Partial/None – 2/1/0)
    5. Example Command:
       1. [Source Address, 0xDE,0x02]
    6. Example Response: No override
       1. [0x60, 0xDE,0x02, 0x00]
43. Request Current Status
    1. Purpose: Query to see current unit settings.
    2. Command: 0xDE, 0x10
    3. Data sent with command (beyond source address and command): NA
    4. Response: General status bytes related to the operation of the water heater. Data returned as follows:
       1. Byte 0 = current heating mode
       2. Byte 1 - # of vacation days remaining in Vacation mode.
       3. Byte 2 = Tank (T2) Temperature in Degrees F
       4. Byte 3 = Setpoint in Degrees F
       5. Bytes 4 = % of filter hours remaining
       6. Byte 5 – Demand response fall back mode
       7. Byte 6 – Demand response fall back temperature
       8. Byte 7 – current user mode
    5. Example Command:
       1. [Source Address, 0xDE,0x10]
    6. Example Response: No example data…
       1. [0x60, 0xDE,0x10, …]
44. Request KWH Data
    1. Purpose: Query energy usage.
    2. Command: 0xDE, 0x12
    3. Data sent with command (beyond source address and command): NA
    4. Response: Response includes instantaneous energy in Watts, #of Watt Seconds used since last query, # of seconds since last query. Data returned as follows:
       1. Byte 0,1 = Instantaneous power Watts
       2. Byte 1-4 = Watt Seconds since last query
       3. Byte 5-8 = # of seconds since last Query
       4. Byte 9 = 0 – fixed for water heater at this time.
    5. Example Command:
       1. [Source Address, 0xDE,0x12]
    6. Example Response: No example data…
       1. [0x60, 0xDE,0x12, …]
45. Request Error Codes
    1. Purpose: Query whether unit has any error codes set.
    2. Command: 0xDE, 0x14
    3. Data sent with command (beyond source address and command): NA
    4. Response: Coded list of error codes
       1. Byte 0 =
          1. Bit0 – DataFlash Fail Status
          2. Bit1 – Stuck Key Fail Status
          3. Bit2 – Dry Tank Fail Status
          4. Bit3 – Miswire Fail Status
          5. Bit4 – 0
          6. Bit5 – 0
          7. Bit6 – 0
          8. Bit7 – 0
       2. Byte 1
          1. Bit0 – UE Fail Status
          2. Bit1 – LE Fail Status
          3. Bit2 – Filter Fail Status
          4. Bit3 – T3A Fail Status
          5. Bit4 – T3B Fail Status
          6. Bit5 – T4 Fail Status
          7. Bit6 –T5 Fail Status
          8. Bit7 – T2 Fail Status
       3. Byte 2
       4. Bit0 - SS Cond A Status
       5. Bit1 - SS Cond B Status
       6. Bit2 - SS Cond C Status
       7. Bit3 - SS Cond D Status
       8. Bit4 – SS Cond E Status
       9. Bit5 – SS Cond F Status
       10. Bit6 – SS Fan Status
       11. Bit7 – CMP Fail Status
    5. Example Command:
       1. [Source Address, 0xDE,0x14]
    6. Example Response: No example data…
       1. [0x60, 0xDE,0x14, 0x##, 0x##, 0x## …]
46. Remote DSM Override
    1. Purpose: Returns unit to consumer mode regardless of override status.
    2. Command: 0xDE, 0x30
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command:
       1. [Source Address, 0xDE,0x30]
    6. Example Response: NA
47. Update Demand Response Level
    1. Purpose: Sets the current Demand pricing level 0 – 3 (low, medium, high, critical)
    2. Command: 0xDF, 0x00
    3. Data sent with command (beyond source address and command): 1 Byte indicating the level 0-3
    4. Response: NA
    5. Example Command: Normal mode set…
       1. [Source Address, 0xDF,0x00, 0x00]
    6. Example Response: NA
48. Clear Energy Usage Counters
    1. Purpose: Clear the accumulated power Watt/Second sum as well as the time since the energy was last queried.
    2. Command: 0xDF, 0x07
    3. Data sent with command (beyond source address and command): NA
    4. Response: NA
    5. Example Command: Normal mode set…
       1. [Source Address, 0xDF,0x07]
    6. Example Response: NA
49. Set Mode Command
    1. Purpose: Update the Current heating mode of the water heater.
    2. Command: 0xDF, 0x14
    3. Data sent with command (beyond source address and command): 3 bytes clarifying the mode as follows:
       1. Byte 1 – Operational mode (0-4)
          1. 0 = Hybrid
          2. 1 = Standard Electric
          3. 2 = Economy Heat
          4. 3 = High Demand
          5. 4 = Vacation
       2. Byte 2 – Vacation Mode Modifier - # of days to remain in Vacation Mode.
       3. Byte 3 – Vacation Mode Modifier – Set Point while on vacation in Degrees F (normally this would be set to 50 or 60 degrees F)
    4. Response: Confirmation that the mode was accepted. Confirmation takes form Success/Mode not Available/Mode Invalid (0/1/2)
    5. Example Command: Enter Economy Heat
       1. [Source Address, 0xDF,0x14, 0x03]
    6. Example Response: Successful mode change.
       1. [0x60, 0xDF,0x14,0x00]