CURTIS

# Manual

# Model **1030**

## Acuity® Battery Monitoring System


**Curtis Instruments, Inc.**
200 Kisco Ave
Mt. Kisco, NY 10549
www.curtisinstruments.com


Read Instructions Carefully!



53069, Rev D April 2025

# TABLE OF CONTENTS

## TABLE OF CONTENTS cont'd

## TABLE OF CONTENTS cont'd

### FIGURES



### TABLES



The Curtis Difference
You feel it when you drive it

# 1 — OVERVIEW

The Curtis Model 1030 Acuity® Battery Monitoring System includes:

- The Model 1030 Curtis Acuity® Module — 17668700-xxx.
- Acu-Set™ 2.0 Software — Acuset.exe.
- CAN-to-USB Dongle — 17697USBCANI-01.
- Acuity Setup Harness — 17668357.

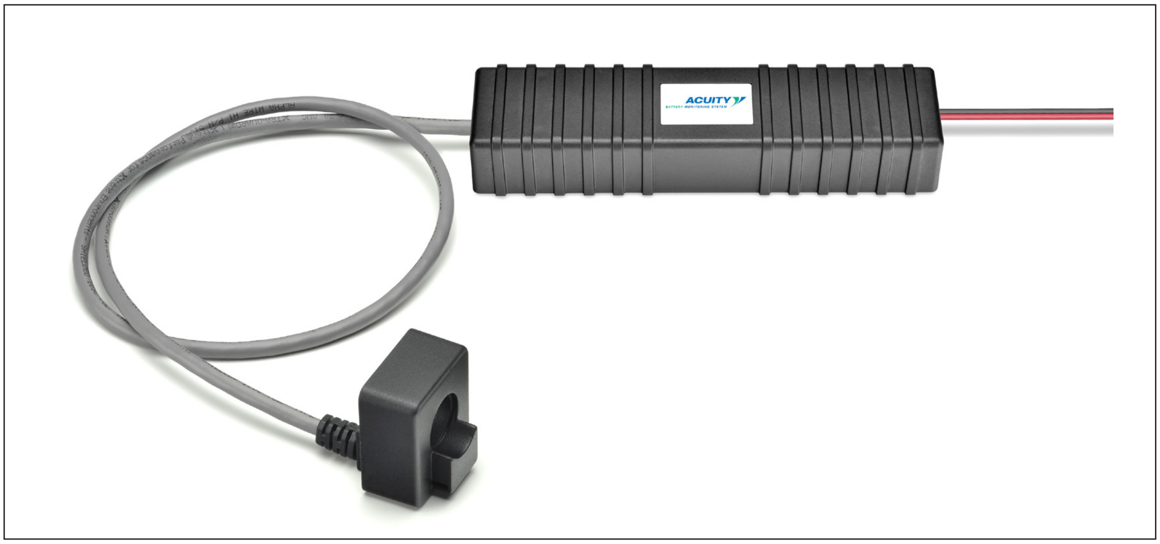


**Figure 1-1**
*Curtis Acuity® Battery Monitoring System.*

The system mounts directly to an industrial vehicle lead acid battery and measures, records, and transmits battery performance data throughout the life of the battery.

The Curtis Acuity is ideal for use in electric vehicles with applications such as material handling, airport ground support, floor cleaning, light-on-road, golf/utility, and aerial work platforms. **Features include:**

- Highly accurate state-of-charge calculation that uses a weighted average of ampere-hour counting and voltage under load measurements.
- Witness data that demonstrates the battery has been operated within the conditions of its warranty.
- Since Acuity is permanently attached to the battery, information is collected consistently and accurately over the lifetime of the battery, no matter how many times the battery is moved.
- Data can used to optimize productivity of a battery fleet/vehicle.
- CANbus allows simple system integration.
- Installation is simple and non-invasive, with no need for special hardware and no cutting of cables or drilling into the battery.

- Integral real-time-clock allows date and time stamping of significant events related to the battery or any vehicle component of the CANbus.
- CANbus isolation eliminates ground loops that can cause component damage as well as data errors due to differences in ground potentials among the nodes on the CANbus.
- By measuring, recording, and communicating battery current, voltage, temperature, and use-time, Acuity can compensate for the effects of variations in load, duty cycle, and operating temperature of any given application.
- Calculates the percent of rated capacity remaining in the battery as an indication of remaining battery life.
- Curtis Acu-Set software, when installed on a computer connected to an Acuity via a CAN-to-USB dongle, allows Acuity to be configured to match the specific battery to which it is mounted.
- Historical data can be uploaded to a PC.
- Instantaneous battery performance data can be viewed on a PC (Power Prover mode).

## CANopen CONVENIENCE

The Acuity is CANopen compliant, responding to the standard NMT, PDO, and SDO communications as well as the DS301-required identity and standard objects. The Curtis CANopen extensions allow additional features, such as OEM and User default configurations.

The Acuity will receive a single SDO and respond with a single SDO. These SDOs are fixed, simplifying the interface to a VCL-enabled device. All programmable parameters and viewable values within the Acuity are accessible via standard SDO transfer.

Battery information is displayed in real time on the Curtis enGage® VII or any other CAN-based display.

*Familiarity with your Curtis Acuity system will help you install and operate it properly. We encourage you to read this manual carefully. If you have questions, please contact your local Curtis office.*

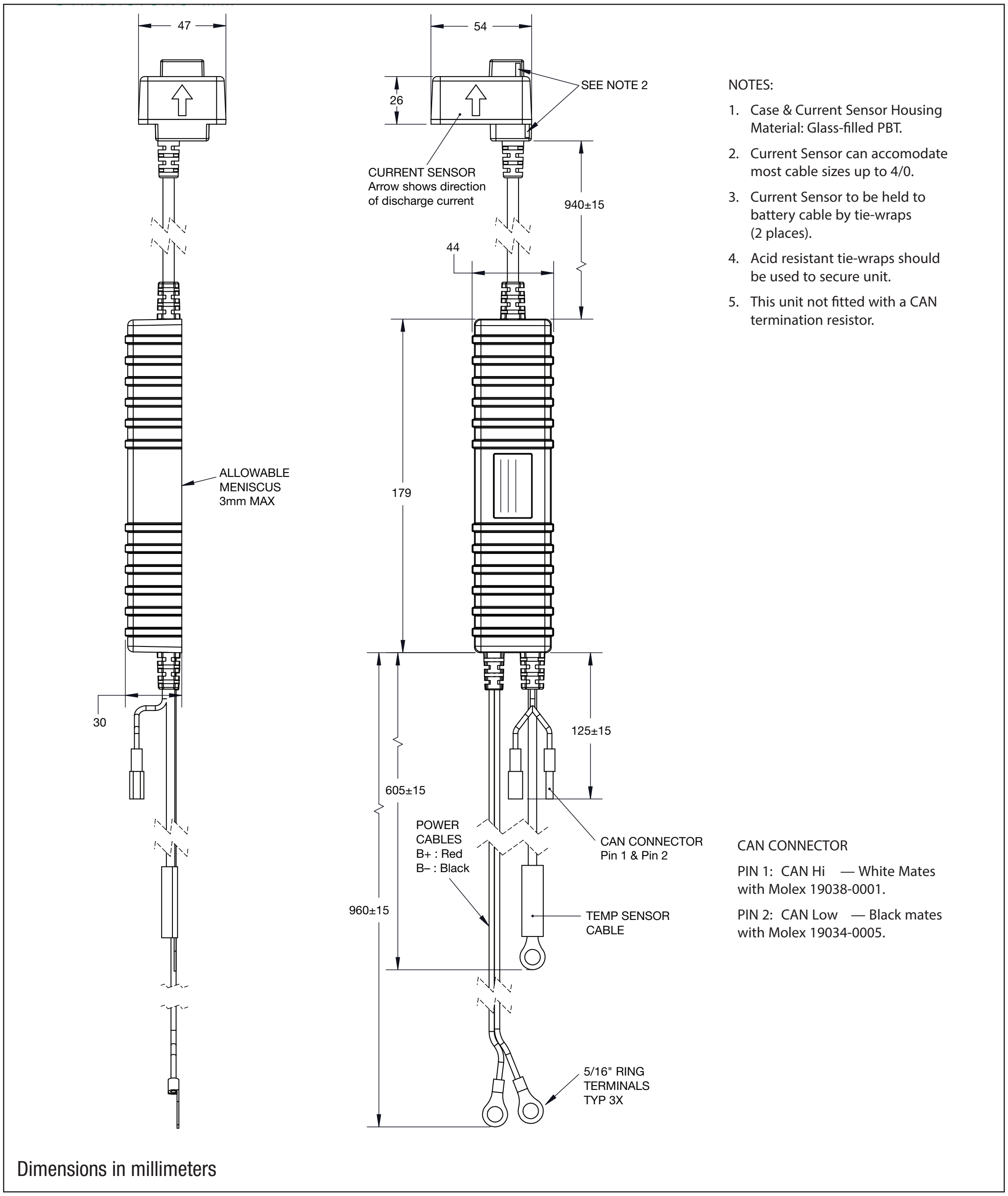
# 2 — INSTALLATION AND WIRING

The outline and dimensions for the 1030 Acuity are shown in Figure 2-1.

**Perform the installation in an area that is well ventilated. Before installing the Acuity module, clean the battery, cables, and terminals.**

**Figure 2-1**

*Mounting dimensions, Curtis 1030 Acuity.*



## MOUNTING THE ACUITY MODULE

Locate the Acuity module on the battery in such a way as to avoid damage to the Acuity through normal battery/vehicle use. Use acid-resistant tie wraps to secure the Acuity to the battery using the intercell connectors and the ribs that are molded into the Acuity module; see Figure 2-2.

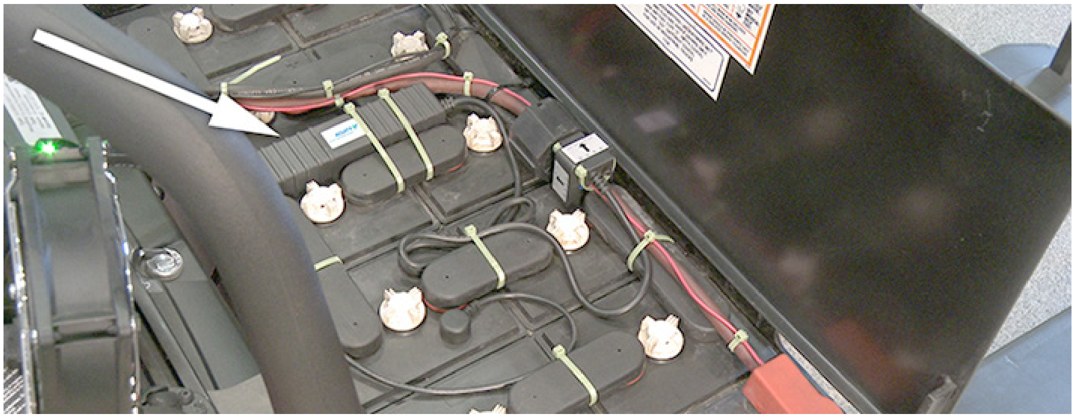


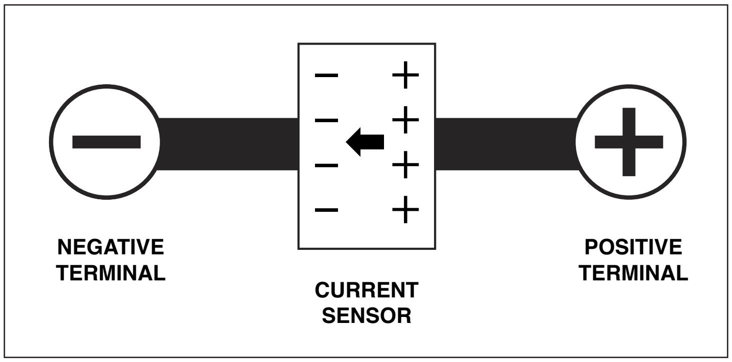
**Figure 2-2**
*Typical installation showing batteries with Acuity installed.*

## CONNECTING B+, B–, AND THE TEMPERATURE SENSOR

The Acuity's B+, B–, and temperature sensor connections are made via 5/16" ring terminals. Simply place the ring terminal over the battery stud/terminal and then replace the washer and nut.

## INSTALLING THE CURRENT SENSOR

The current sensor is polarity sensitive and must be properly oriented for the Acuity to work correctly.




Either B+ or B– can be selected to pass through the current sensor, depending on the battery configuration and location of the current sensor.

Pass the end of the battery cable through the opening of the current sensor, making sure that the marking on the current sensor is pointing in the correct direction.

Insert a tie wrap into each of the slots in the current sensor and pull the battery cable tight against the side of the current sensor in which the slots are located.

## CAN CONNECTIONS

The CAN connector is either a 2-pin Deutsch or two bullet-style connectors. Note that the Deutsch connector is not acid resistant, and therefore if used should be located at least one meter away from the battery.

The termination resistors on a cable should match the nominal impedance of the cable. ISO 11898 requires a cable with a nominal impedance of 120 Ω; therefore, you should use 120 Ω resistors for termination. If you place multiple devices along the cable, only devices at the ends of the cable need termination. You can specify whether the Acuity will include the 120 Ω termination resistor (see Specifications).

# 3 — ACU-SET SOFTWARE

Acu-Set software is license-based and will only run on the PC on which it is originally installed. After the software is purchased, instructions will be emailed from Curtis to define the download and activation procedure. The license for Acu-set can only be used on one computer at a time. For information on releasing a license so that it can be installed on another computer, see Managing the License.

A license expires in 120 days if the computer remains offline. After the computer is reconnected to the internet, the license will automatically renew for another 120 days.

## CONNECTING THE ACUITY TO THE COMPUTER

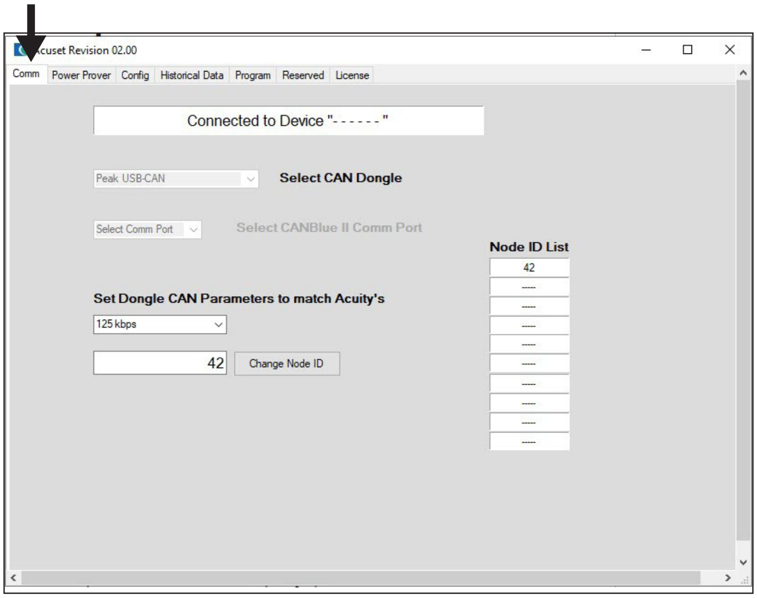
Make sure the Acuity is powered on before making the connections, then follow these steps.

- Using the Peak USB-to-CAN dongle (Curtis p/n 17697USBCANI-01) and mating harness (Curtis p/n 17668357):
  - Connect bullet connectors on Acuity to mating bullet connectors on harness
  - Connect sub-D connector on harness to mating sub-D connector on Peak dongle.
- Install the dongle USB driver by inserting the CD into the computer and following the instructions.
- Copy the PCANBasic.dll file for the Peak dongle from the dongle CD to the folder where the Acu-Set software is executable (the folder where the .exe file is located). The process is the same for the other brands of dongle: i.e., the driver needs to be downloaded and the .dll file must be in the same folder as the Acu-Set executable.
- Plug the dongle's USB connector into the USB slot on the computer.
- From within the folder in which the Acu-Set files were saved, double-click on the Acu-Set.exe file.

## ESTABLISHING COMMUNICATION BETWEEN ACUITY AND COMPUTER

Follow these steps to establish communication.

- Click on the COMM tab at the top of the Acu-Set screen.




- From the Select CAN Dongle drop-down menu, select one of the following CAN dongles:
  - Kvaser BlackBird WiFi
  - SYS TEC USB-CAN
  - Peak USB-CAN [default selection]
  - CANBlue II.

  **Note:** If you are using the CANBlue Bluetooth dongle, select `CANBlue # Comm Port` in the field below the dongle selection field.

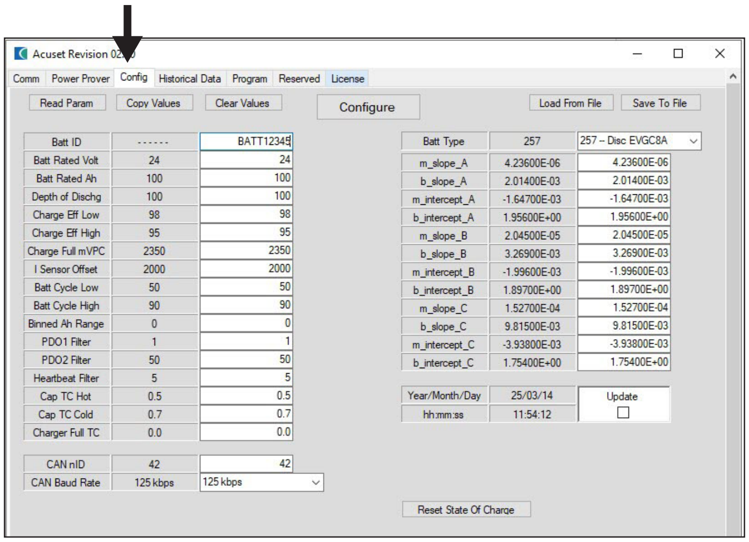
- When the default Peak dongle is chosen and the baud rate and node ID match the Acuity, Acu-Set will automatically display the Power Prover tab. If this did not happen, review the Baud Rate and CAN Node ID settings.

  The default baud rate is 125 kbps. If it is necessary to change this setting to match your system, select the proper baud rate from the drop down menu:
  - 100 kbps
  - 125 kbps [default setting]
  - 250 kbps
  - 500 kbps
  - 1000 kbps.

The default Node ID is 42. If it is necessary to change this value, enter a value between 1 and 127 that is not already in use. The Node ID List shows the Node IDs of all devices that are transmitting in that network at the selected baud rate.

## CONFIGURING THE ACUITY

The Acuity needs to be configured to match the battery on which it is installed. Begin by clicking on the Config tab at the top of the screen.




There are three methods of configuration.

1. Click on the pull-down menu entitled Batt Type and select the battery type that matches your battery. Enter ID (Batt ID), voltage (Batt Rated Volt), capacity (Batt Rated Ah) and empty point (Depth of Disch). Click **Configure**, then click **Reset State of Charge**.

   **Note:** For depth of discharge, if desired empty point is 20% state-of-charge, enter 80. See explanation on page 22.

   A dialog box will open to indicate that the Acuity has been configured successfully. Click **OK** and the parameters of that configuration will be loaded into the column on the left.

   **Note:** The table on the right side of the screen is used for internal purposes only.

2. A configuration file that has been previously stored on the computer can be retrieved and programmed into the Acuity.

   Click on **Load from File**, select the appropriate configuration file, and click **Open**. Configuration files use the extension .Acfg.

   The parameters of that configuration will be loaded into the right-hand column of the table on the left side of the screen. Click on **Configure** button.

   A dialog box will open to indicate that the Acuity has been configured successfully. Click **OK**.

3. Use the other fields in the Config tab.

   **Read Param** Reads all parameters from the Acuity and displays them into the column on the left of the table. This feature allows the user to determine how an existing Acuity is configured when newly connected to the computer.

   **Copy Values** This function is used when copying parameters from an Acuity to a computer. After the parameters are read from the Acuity, the **Copy Values** button activates the copying of the values on the left to the editable list on the right. The user can then make any necessary changes before **Save to File**.

**Clear Values** This function removes all the values that were entered by the user.

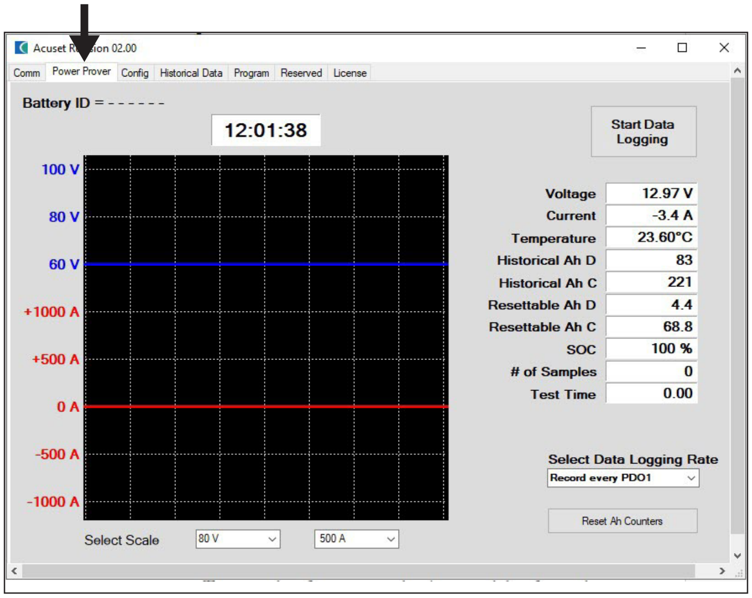
**Save to File** Saves the parameters that have been entered into the editable list on the right.

**Reset State-of-Charge** Resets that value in the Acuity.

**Update Time & Date** Allows the user to set the time and date of the real-time clock in the Acuity.

## POWER PROVER (VIEWING & RECORDING INSTANTANEOUS DATA)

The Power Prover screen allows the user to view live data being transmitted from the Acuity and to record that data on the host computer.




The Power Prover screen contains the following functions.

**Select Scale** Two pull-down menus allow you to select the voltage scale and the current scale to be displayed on the computer.

**Select Data Logging Rate** The rate at which data is recorded to a file on the computer can be selected using the drop-down menu. It is set in multiples of PDO1 messages. The PDO1 transmission rate is 100 ms by default. It can be reset using a CAN object.

**Start Data Logging** Begins storing data that is received from the Acuity's measurements into a file on the computer, at the rate set by **Select Data Logging Rate**.

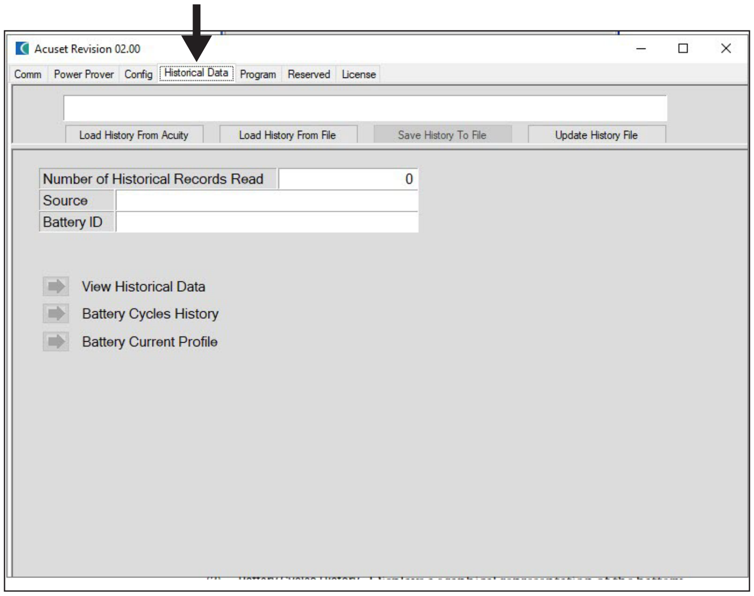
**Reset AHr Counters** Resets the Ampere Hour counters to zero. "D" represents Ampere Hours <u>D</u>ischarged, and "C" represents Ampere Hours <u>C</u>harged.

**Battery ID** This is an identifier unique to that Acuity and therefore to that battery.

**Time-of-Day** Time of day as reported by the host computer.

## HANDLING HISTORICAL DATA

The Historical Data screen allows the user to retrieve historical data from the Acuity.




Four buttons near the top allow the user to handle the historical data.

**Load History From Acuity** This function retrieves data records from the memory of the Acuity. The number of records and the unique identifier of the Acuity (and this battery) are displayed. There are three ways to view the data, each with its own subscreen.

1. **View Historical Data** Displays the raw data for each Acuity parameter in tabular form. The battery parameters plotted using **Select Parameters to Plot** are:

   State of Charge

   Temperature

   Ah Delivered / Returned

   Percent Ah Returned

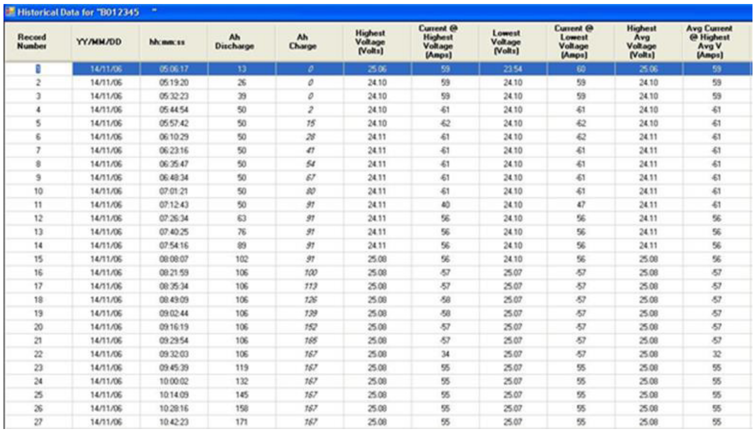
   Estimated Ah Capacity.

2. **Battery Cycles History** Displays a graphical representation of the battery history data. Slide bars help select specific ranges of data to view and simplify the identification of measurements of particular interest, such as when the state of charge has fallen below 20%.

3. **Battery Current Profile** Displays a bargraph showing the ranges over which the ampere hours were consumed.

Examples of these three data displays are shown on the next page.

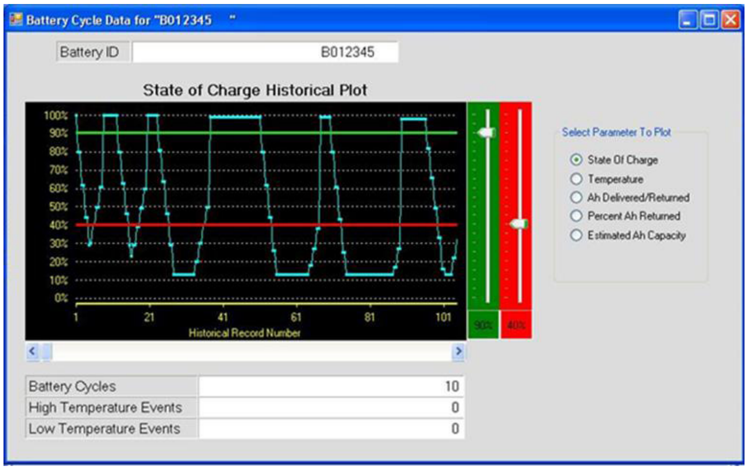
## 1 View Historical Data



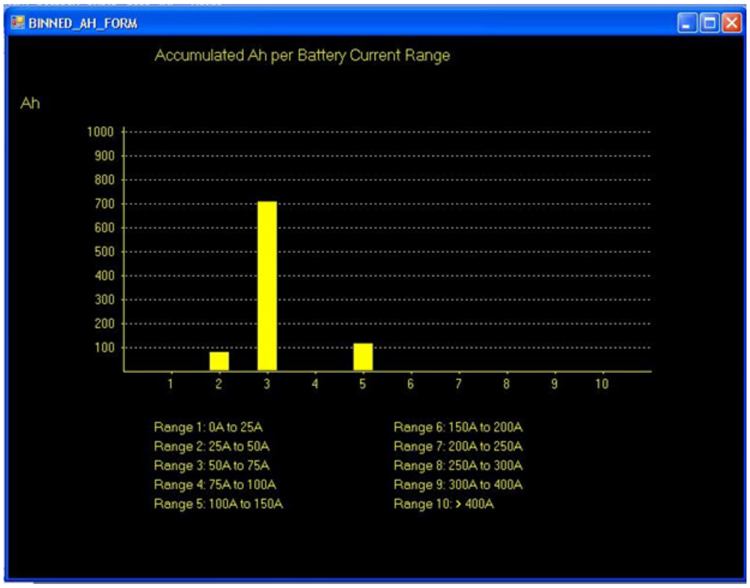
Historical Data for "B012345 "

| Record Number | YY/MM/DD | hh:mm:ss | Ah Discharge | Ah Charge | Highest Voltage (Volts) | Current @ Highest Voltage (Amps) | Lowest Voltage (Volts) | Current @ Lowest Voltage (Amps) | Highest Avg Voltage (Volts) | Avg Current @ Highest Avg V (Amps) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 14/11/06 | 05:06:17 | 13 | 0 | 25.06 | 59 | 23.54 | 60 | 25.06 | 59 |
| 2 | 14/11/06 | 05:19:20 | 26 | 0 | 24.10 | 59 | 24.10 | 59 | 24.10 | 59 |
| 3 | 14/11/06 | 05:32:23 | 39 | 0 | 24.10 | 59 | 24.10 | 59 | 24.10 | 59 |
| 4 | 14/11/06 | 05:44:54 | 50 | 2 | 24.10 | -61 | 24.10 | -61 | 24.10 | -61 |
| 5 | 14/11/06 | 05:57:42 | 50 | 15 | 24.10 | -62 | 24.10 | -62 | 24.10 | -61 |
| 6 | 14/11/06 | 06:10:29 | 50 | 28 | 24.11 | -61 | 24.10 | -62 | 24.11 | -61 |
| 7 | 14/11/06 | 06:23:16 | 50 | 41 | 24.11 | -61 | 24.10 | -61 | 24.11 | -61 |
| 8 | 14/11/06 | 06:35:47 | 50 | 54 | 24.11 | -61 | 24.10 | -61 | 24.11 | -61 |
| 9 | 14/11/06 | 06:48:34 | 50 | 67 | 24.11 | -61 | 24.10 | -61 | 24.11 | -61 |
| 10 | 14/11/06 | 07:01:21 | 50 | 80 | 24.11 | -61 | 24.10 | -61 | 24.11 | -61 |
| 11 | 14/11/06 | 07:12:43 | 50 | 91 | 24.11 | 40 | 24.10 | 47 | 24.11 | -61 |
| 12 | 14/11/06 | 07:26:34 | 63 | 91 | 24.11 | 56 | 24.10 | 56 | 24.11 | 56 |
| 13 | 14/11/06 | 07:40:25 | 76 | 91 | 24.11 | 56 | 24.10 | 56 | 24.11 | 56 |
| 14 | 14/11/06 | 07:54:16 | 89 | 91 | 24.11 | 56 | 24.10 | 56 | 24.11 | 56 |
| 15 | 14/11/06 | 08:08:07 | 102 | 91 | 25.08 | 56 | 24.10 | 56 | 25.08 | 56 |
| 16 | 14/11/06 | 08:21:59 | 106 | 100 | 25.08 | -57 | 25.07 | -57 | 25.08 | -57 |
| 17 | 14/11/06 | 08:35:34 | 106 | 113 | 25.08 | -57 | 25.07 | -57 | 25.08 | -57 |
| 18 | 14/11/06 | 08:49:09 | 106 | 126 | 25.08 | -58 | 25.07 | -57 | 25.08 | -57 |
| 19 | 14/11/06 | 09:02:44 | 106 | 139 | 25.08 | -58 | 25.07 | -57 | 25.08 | -57 |
| 20 | 14/11/06 | 09:16:19 | 106 | 152 | 25.08 | -57 | 25.07 | -57 | 25.08 | -57 |
| 21 | 14/11/06 | 09:29:54 | 106 | 165 | 25.08 | -57 | 25.07 | -57 | 25.08 | -57 |
| 22 | 14/11/06 | 09:32:03 | 106 | 167 | 25.08 | 34 | 25.07 | -57 | 25.08 | 32 |
| 23 | 14/11/06 | 09:45:39 | 119 | 167 | 25.08 | 55 | 25.07 | 55 | 25.08 | 55 |
| 24 | 14/11/06 | 10:00:02 | 132 | 167 | 25.08 | 55 | 25.07 | 55 | 25.08 | 55 |
| 25 | 14/11/06 | 10:14:09 | 145 | 167 | 25.08 | 55 | 25.07 | 55 | 25.08 | 55 |
| 26 | 14/11/06 | 10:28:16 | 158 | 167 | 25.08 | 55 | 25.07 | 55 | 25.08 | 55 |
| 27 | 14/11/06 | 10:42:23 | 171 | 167 | 25.08 | 55 | 25.07 | 55 | 25.08 | 55 |

## 2 Battery Cycles History




## 3 Battery Current Profile



Here are the remaining buttons on the Historical Data screen.

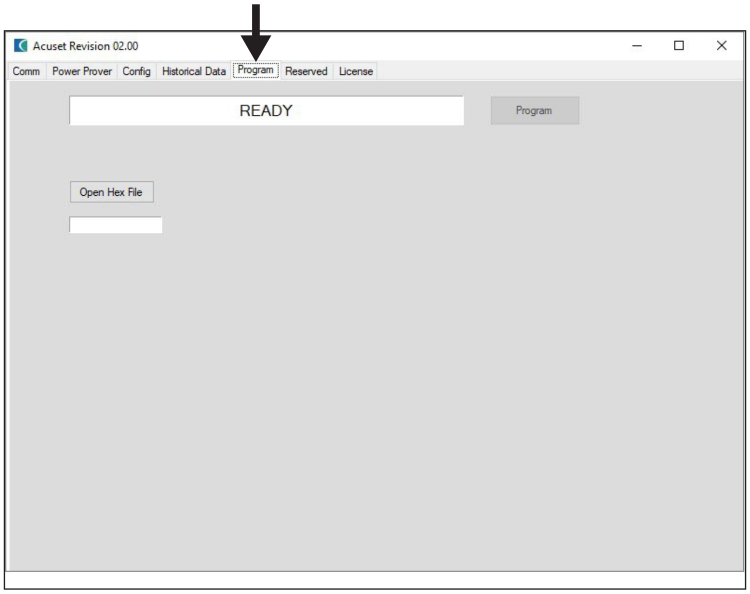
**Load History From File** Retrieves data records previously stored in the computer and displays the data in tabular form. The file format extension is .*AHR.

**Save History To File** Stores data records from Acuity's memory to a file on the computer in .ahr format.

**Update History File** Updates stored history file with new data. This function also provides an option for creating a backup for the stored history file before updating.

## PROGRAMMING THE ACUITY

The Program screen is used to re-program the Acuity with updated firmware.

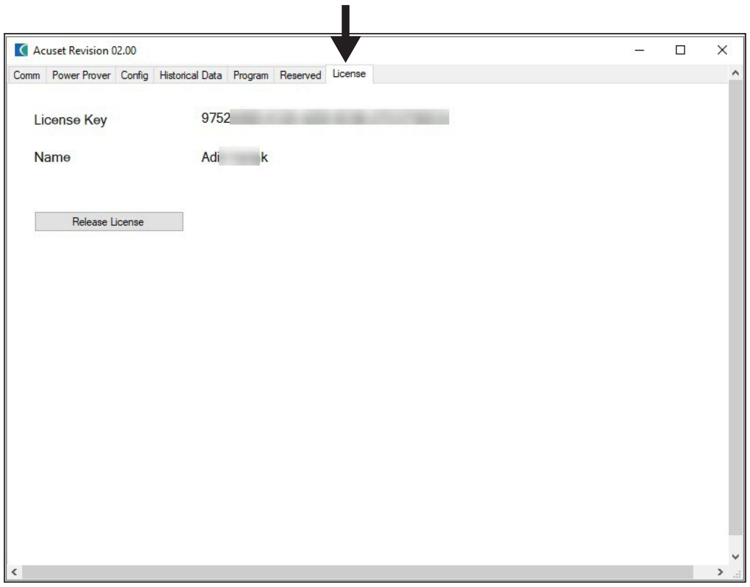



**Open Hex File** Retrieves the hex file from the computer's folder.

**Program** Downloads the file to the Acuity.

## MANAGING THE LICENSE

The License tab indicates the name of the licensed user and the license key. The Release License button is for releasing the license so that it can be used on another computer.



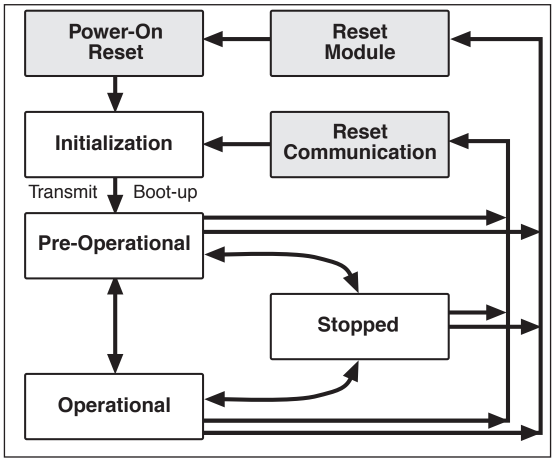

**Note:** If the computer on which the license is installed is dead, ask Curtis technical support to release the license.

# 4 — CANOPEN COMMUNICATIONS

The Acuity adheres to the industry standard CANopen communication protocol and thus will easily connect into many CAN systems, including those using the Curtis AC and Vehicle System controllers (1234/36/38, 1298, 1310, and enGage VII). Any CANopen-compatible commander node can be programmed to control the Acuity.

## MINIMUM STATE MACHINE

The Acuity will run the CANopen minimum state machine as defined by CiA. The CANopen minimum state machine has four defined states: Initialization, Pre-Operational, Operational, and Stopped.




When the Acuity powers up, it goes to the Initialization state; this is also known as the Boot-up state. No CAN communications from the Acuity are transmitted in this state although the Acuity listens to the CANbus. When the Acuity has completed its startup and self-tests, it issues an initialization heartbeat message and automatically goes to the Pre-Operational state and the to the Operational state.

In the Operational state, the Acuity will start sending PDOs and process all other necessary CANopen messages.

## BAUD RATES

The Acuity will run at one of five selectable baud rates: 125 kbps, 250 kbps, 500 kbps, 800 kbps, and 1 Mbps. Rates below 125 kbps are not supported.

The baud rate can be changed by an SDO. Changes in the baud rate require an NMT reset to make the new rate active.

## NODE ADDRESSES

The node address of the  kbps can be 1 to 127 and is used by CANopen to route messages to the Acuity and to denote messages *from* the Acuity. The node address is part of the COB-ID and therefore also plays a part in message priority and bus arbitration.

Changes to the node address require an NMT reset or power-cycle.

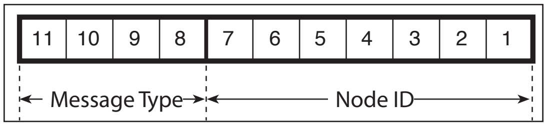
## STANDARD MESSAGE IDENTIFIERS

The Acuity will produce—and respond to—the standard message types with the following CANopen identifiers.

| Message Type | Message Identifier |
|---|---|
| NMT | 0000 – 00h |
| EMERGENCY | 0001 – 01h |
| PDO-TRANSMIT | 0011 – 03h |
| PDO-RESPONSE | 0100 – 04h |
| SDO-TRANSMIT | 1011 – 0Bh |
| SDO-RESPONSE | 1100 – 0Ch |
| HEARTBEAT | 1110 – 0Eh |

The 11-bit identification field is a fixed part of the CANopen specification called the **C**ommunication **OB**ject **ID**entification (COB-ID). This field is used for arbitration on the bus. The COB-ID with the lowest value gets priority and wins arbitration. Consequently, NMT messages have the highest priority of the standard message types, and the heartbeat has the lowest priority.

The standard organization of the COB-ID puts the message type in the upper four bits, and the Node ID in the bottom seven bits:



## NMT MESSAGES

NMT (**N**etwork **M**anagement **T**ransmission) messages are the highest priority message available. The NMT message puts the Acuity into one of the four defined states. These messages have 1 byte of data sent by the commander; the responder does not respond with any data to an NMT. The Acuity state value is transmitted with each heartbeat message.

| Value | State |
|---|---|
| 00h | Initialization (or "boot-up") |
| 04h | Stopped |
| 05h | Operational |
| 7Fh | Pre-Operational |

The NMT message identifier consists of the standard message type (NMT) in the top four bits; the bottom seven bits must be set to zero.

The first data byte of the NMT command is the command specifier:

| Value | Command Specifier |
|---|---|
| 01h | Enter the Operational state |
| 02h | Enter the Stopped state |
| 80h | Enter the Pre-Operational state |
| 81h | Reset Acuity (warm boot) |
| 82h | Reset the CANbus |

The second byte of the NMT command defines whether this NMT is for all responder nodes on the bus (data byte = 00h) or for a specific node (data byte = Node ID of the Acuity).

## HEARTBEAT MESSAGES

The heartbeat message is a very low priority message, periodically sent by each responder device on the bus. The heartbeat message has a single byte of data and requires no response. Once the Acuity is in the Pre-Operational state, the next heartbeat will be issued and will continue until communication is stopped.

The heartbeat message has only one data byte. The top bit is reserved and should be set to zero. The bottom 7 bits hold the current NMT device state as defined previously.

# 5 — PDO COMMUNICATIONS

The Curtis Acuity is easily controlled and monitored through two fixed communication packets. Each data packet contains 8 bytes. CANopen calls these packets **P**rocess **D**ata **O**bjects (PDOs). PDO messages have a medium priority.

The PDO communication packets conserve bus bandwidth by bundling the values of a group of objects into a single message. The content of these PDOs is fixed, thus simplifying the interface.

The Acuity transmits PDO1, PDO2, and PDO4 continuously. By default, PDO1 is sent every 100 ms, PDO2 is sent every 5 seconds, and PDO4 is sent every second. PDO1 and PDO2 transmit periods are configurable using a CAN object.

## PDO1

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| Battery Voltage LSB | Battery Voltage MSB | Battery Current LSB | Battery Current MSB | Temperature LSB | Temperature MSB | Not Used | SoC |

**Battery Voltage** Unsigned 16-bit integer. Resolution is hundredths of volts.

**Example:** A value of 30000 equals 300.00 V.

**Battery Current** Signed 16-bit integer. Positive value represents current coming out of battery (discharge). Units are in tenths of amperes.

**Example:** A value of +3456 equals 345.6 A of discharge current.

**Temperature** Signed 16-bit integer. Units are in hundredths of degrees Celcius.

**Example:** A value of 5500 equals 55.00°C.

**SoC (State of Charge)** Unsigned 8-bit integer. Range is 0 – 100%.

## PDO2

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| Historical Discharge Ah LSB | Historical Discharge Ah | Historical Discharge Ah | Historical Discharge Ah MSB | Historical Charge Ah LSB | Historical Charge Ah | Historical Charge Ah | Historical Charge Ah MSB |

## PDO4

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| Seconds | Minutes | Hours | Day | Month | Year | [Reserved] | [Reserved] |

# 6 — SDO COMMUNICATIONS

CANopen uses **S**ervice **D**ata **O**bjects (SDOs) to change and view all internal parameters, or "objects." The SDO is an 8-byte packet that contains the address and sub-address of the parameter in question, whether to read or write that parameter, and the parameter data (if it is a write command). SDOs are sent infrequently and have a low priority on the CANbus.

SDOs are designed for sporadic and occasional use during normal runtime operation. There are two types of SDOs: expedited and block transfer. The Acuity does not support large file uploads or downloads (using the block transfer), so all SDOs in this specification are expedited SDOs.

The SDOs in the Acuity are used to set up and input battery-specific parameters. They are also used to retrieve basic information (such as version or battery-specific data).

## SDO COMMANDER REQUEST

An SDO transfer always starts with a request message from the commander node. Each SDO request message consists of one control byte, a two-byte CAN Object index, a one-byte CAN Object sub-index, and up to 4 bytes of valid data. This format is CANopen compliant.

**SDO Commander Request** *(received from the system commander)*

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| Control | CAN Object Index | | Sub-index | Data | Data | Data | Data |

The first data byte contains R/W message control information.

| Action | Byte 1 Value |
|---|---|
| Read | 42h |
| Write | 22h |

The next two data bytes hold the CAN Object index. The least significant byte of the index appears first, in byte 2, and the most significant byte appears in byte 3. For example, if the index is 3021h, byte 2 holds the 21h and byte 3 holds the 30h.

Data byte 4 holds the CAN Object sub-index. When there is only one instance of a parameter or value type, this value is 0. If there are several related parameters or values, the sub-index is used.

The last four data bytes hold the data that is to be transferred. In the case of a single-byte transfer, the data is placed into data byte 5, with bytes 6 through 8 being undefined (set to 0). In the case of a 16-bit transfer, the lower 8 bits appear in data byte 5 and the upper 8 bits appear in data byte 6; bytes 7 and 8 are undefined (set to 0). The case of a 32-bit transfer follows the same strategy, with the least significant byte placed in data byte 5 and the most significant byte placed in data byte 8.

## SDO RESPONSE

An SDO request is always acknowledged with a response message from the Acuity. The Acuity can issue two kinds of response messages: a normal response or, in case of an error in the request SDO, an Abort SDO Transfer message.

**SDO Response** *(sent by the Acuity in response to the system commander)*

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| Control | CAN Object Index | | Sub-index | Data: either the requested Read values, or the actual Write values, or an error code | | | |

The first data byte of the response contains an acknowledge code, which depends on the type of transfer that was initially requested.

| Action | Byte 1 Value |
|---|---|
| Read Response | 40h |
| Write Acknowledge | 60h |
| Abort SDO | 80h |

Data bytes 2, 3, and 4 hold the CAN Object index and sub-index of the request SDO.

If the SDO was a read command (a request for data from the Acuity), data bytes 5 through 8 will be filled with the requested values, with the least significant byte is data byte 5 and the next least significant in byte 6 and so forth. All unused bytes are set to 0.

If the SDO was a write command, data bytes 5 through 8 will return back the **actual** value written in bytes 5 – 8. In this way, if the Acuity needs to limit or round-down the SDO write request, the commander will know—because the return value will be different than the sent value.

If the SDO did not properly read or tried to access a parameter improperly, an Abort SDO Transfer will be sent. Data bytes 5 through 8 will be filled with a 32-bit error code:

06020000h = Object does not exist
06010002h = Attempt to write to a read only object.

## TYPES OF SDO OBJECTS

There are three types of SDO objects: *Communications Profile Objects* (address range 1000h), *Device Parameter Objects* (address range 5100h), and *Device Monitor Objects* (address range 5200h). Communications Profile Objects are described in this section; Device Parameter Objects and Device Monitor Objects have sections of their own. The following definitions apply to all the objects.

**Access** RO = Read Only access. RW = Read/Write access.

**Index** The CAN address that is used to access this object.

**Sub-index** Some objects have several associated values. In these cases, a sub-index is used to access each component of the object.

## COMMUNICATION PROFILE OBJECTS

The objects found in the 1000h CAN Object address range are shown below in Table 6-1.

**Table 6-1 Communication Profile Objects**

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 1000h | 00h | RO | 0 | 0 | 0 | 0 | Device Type. |
| 1001h | 00h | RO | 0 | 0 | 0 | 0 | Error Register. |
| 1008h | 00h | RO | 1 | 0 | 3 | 0 | Model Number.<br>Four ascii characters; = 1030. |
| 1009h | 00h | RO | 0 | 0 | 0 | 1 | Hardware Version.<br>Four ascii characters. |
| 100Ah | 00h | RO | 0 | 0 | 0 | 1 | Software Version.<br>Four ascii characters. |
| 101Bh | 00h | RO | 5 | 0 | 0 | 0 | Identity Object.<br>Number of entries = 5. |
| | 01h | RO | 49h | 43h | 00h | 00h | Identity Object.<br>Vendor ID = 00004349h. |
| | 02h | RO | 1 | 0 | 3 | 0 | Identity Object.<br>Model Number; four ascii characters. |
| | 03h | RO | 0 | 0 | 0 | 1 | Identity Object.<br>Model Number sequential code;<br>Four ascii characters. |
| | 04h | RO | 0 | 0 | 0 | 0 | Identity Object.<br>Serial Number. |

# 7 — DEVICE PARAMETER OBJECTS

SDOs can be used to set up battery-specific parameters such as battery system voltage and rated capacity. They can also be used to set up battery operation related parameters such battery depth of discharge, charge full voltage, etc.

The objects found in the 5100h CAN Object address range are shown in Table 7-1. Definitions follow the table.

Table 7-1 Device Parameter Objects

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5100h | 00h | RW | LSB | MSB | ‘B’ | ‘V’ | Battery Voltage. 16-bit value. Keyword: BV |
| 5101h | 00h | RW | LSB | MSB | ‘A’ | ‘h’ | Manufacturer Ah Rating, at 5hr or 6hr. Keyword: Ah. |
| 5102h | 00h | RW | LSB | MSB | ‘T’ | ‘y’ | Battery Profile parameters. Keyword: Ty. |
| 5103h | 00h | RW | # | # | # | # | #### = Battery ID bytes 12..9. |
| | 01h | RW | # | # | # | # | #### = Battery ID bytes 8..5. |
| | 02h | RW | # | # | # | # | #### = Battery ID bytes 4..1. |
| 5105h | 00h | RW | value | ‘D’ | ‘O’ | ‘D’ | Depth of Discharge. |
| 5106h | 00h | RW | #1 | #2 | ‘C’ | ‘E’ | #1 = Charge Efficiency at high rates.<br>#2 = Charge Efficiency at low rates. |
| 5107h | 00h | RW | LSB | MSB | ‘C’ | ‘F’ | Charge Full, in mVPC. |
| 5108h | 00h | RW | LSB | MSB | ‘S’ | ‘O’ | Current Sensor Offset, in mA. |
| 5109h | 00..11h | RW | | | | | Custom discharge parameters; *see Definitions.* |
| 510Ah | 00h | RW | BC_Lth | BC_Hth | ‘B’ | ‘C’ | Battery Cycle Thresholds, low and high. Range 0–100. |
| 510Bh | 00h | RW | HRint | ‘H’ | ‘R’ | ‘I’ | Historical Record store interval; range 0 –3.<br>0 = 1/8C,<br>1 = 1/16C,<br>2 = 1/24C,<br>3 = 1/32C. |
| 510Ch | 00h | RW | CAN Prcl | ‘C’ | ‘P’ | ‘L’ | CAN protocol for V.I.T. time; range 0 –2.<br>0 = CANopen PDO.<br>1 = J1939 proprietary.<br>2 = 3100R. |
| 5111h | 00..15 | RW | 0 | 0 | 0 | 0 | Customer notes. Total of 64 bytes. Addressable 4 bytes at a time, using sub-indexes 1–15. |
| 5112h | 00h | RW | pdo1 | pdo2 | heartB | ‘F’ | PDO1, PDO2, and Heartbeat Tx frequency. Resolution 100 ms. Value of 0 disables PDO. |
| 5113h | 00h | RW | ‘R’ | ‘n’ | ‘g’ | # | Sets binned Ah range. For low range, set Byte8 to 0h. For high range, set Byte8 to 1h. |

**Table 7-1 Device Parameter Objects, cont'd**

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5114h | 00h | RW | sec | min | hour | 'T' | Time of day. |
| | 01h | RW | day | month | year | 'D' | Date. |
| | 02h | RW | DoW | 0 | 0 | 'W' | Day of week. (Sunday = 1) |
| 5115h | 00h | RW | CAL | 'C' | 'T' | 'D' | Calibrate time of day. |
| 5116h | 00h | RW | CapTC_H | CapTC_C | ChgTC | 'c' | Capacity Temp Coefficient, hot and cold.<br>Resolution 0.1% per °C.<br>Charger Full Temp Coefficient,<br>resolution 0.1 mV per °C. |

## DEFINITIONS

**Battery Voltage (5100h)** Battery system voltage for the application.

**Battery Profile Parameters (5102h)** Selects a battery discharge profile from the existing battery discharge profiles, including:

- 0: Flooded, Enersys E85
- 1: Flooded, Enersys E100
- 2: Flooded, Enersys E110
- 3: Flooded, Enersys E125
- 4: Flooded, Enersys E140
- 5: Flooded, Enersys E155
- 6: Flooded, Trojan T105
- 7: Flooded, Trojan T890
- 256: AGM, US Battery AGM-185
- 257: AGM, Discovery EVGC8A-A
- 512: Gel, MK Batteries M24
- 65000: Custom parameters.

**Battery ID (5103h)** Battery can be assigned an identification number consisting of up 12 characters.

**Depth of Discharge (DoD) (5105h)** The DoD value scales the SoC displayed.

**Example 1:** When DoD=80, the unit will report 0% SoC when 20% is left.
**Example 2:** When DoD=100, the unit will report 0% SoC when 0% is left.

The factory default is 80.

**Charge Efficiency (5106h)** The range for this parameter is 0 –100. A setting of 100 means 100% of the energy put in goes into charging, and none is wasted.

**Byte 5** Charge efficiency at higher charge rates; factory default=95.
**Byte 6** Charge efficiency at lower charge rates; factory default=100.

High and low rates are determined by the Ampere-Hour Law.

**Custom discharge parameters (5109h)** If the battery discharge profile for the battery used in the application is not available, enter 65000 (Custom parameters) in 5102h and then create a custom discharge profile in 5109h.

**Battery Cycle Low and High Thresholds (BC_LTh, BC_HTh) (510Ah)** Battery discharge cycles in any application are irregular. BC_LTh and BC_HTh set thresholds to define a battery cycle. The Acuity increments the battery cycle if the battery's SoC drops below BC_LTh and then rises about BC_HTh.

**Historical Record Store Interval (HRint) (510Bh)** This parameter defines the fraction of the rated capacity (discharged or charged) at which the historical record is stored. Historical record storage can be set at 1/8th, 1/16th, 1/24th, or 1/32nd of rated capacity C.

## CONFIGURING PARAMETERS

The Device Parameter Objects can be used to configure the Acuity parameters for the application battery.

Most configurable parameters require a keyword to be sent along with the value to be changed. Keywords consist of one or more ASCII characters. Letters are case sensitive. To permanently store configurable parameters into non-volatile memory, a Store command (keyword STOR) needs to be sent. The timeout between setting the parameters and sending the Store command is 15 seconds.

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 51FFh | 00h | WO | ‘S’ | ‘T’ | ‘O’ | ‘R’ | Store configuration parameters. |

**Example:**

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5100h | 00h | RW | 0x30 | 0 | ‘B’ | ‘V’ | Configures the unit for a 48V battery. |
| 5101h | 00h | RW | 00h | 02h | ‘A’ | ‘h’ | Configures the unit to 512 Ahr. |
| 51FFh | 00h | WO | ‘S’ | ‘T’ | ‘O’ | ‘R’ | Causes these new parameters to be permanently stored in non-volatile memory. |

## RESETTING THE SoC

The following object is used to reset the SoC of the battery to 100%.

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 4000h | 00h | WO | ‘R’ | ‘S’ | ‘T’ | 0 | Resets state of charge to 100%. Byte 8 is a zero. |

# 8 — DEVICE MONITOR OBJECTS

The Device Monitor Objects are found in the 5200h CAN Object address range, as shown in Table 8-1.

The Acuity monitors and records various battery parameters. To retrieve battery data from the Acuity, you can use either PDOs or the objects in Table 8-1.

Table 8-1 Device Monitor Objects

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5200h | 00h | RO | | | | | Voltage. |
| 5201h | 00h | RO | | | | | Current. |
| 5202h | 00h | RO | | | | | Temperature. |
| 5203h | 00h | RO | | | | | SoC. |
| 5204h | 00h | RO | | | | | Historical Ah Discharge. |
| 5205h | 00h | RO | | | | | Historical Ah Charge. |
| 5209h | 00...9h | RO | | | | | Binned Ah. |
| 520Ah | 00h | RO | RmAh_lsb | RmAh_msb | RAh_LSB | RAh_MSB | Resettable Ah counter, Discharge. |
| 520Bh | 00h | RO | RmAh_lsb | RmAh_msb | RAh_LSB | RAh_MSB | Resettable Ah counter, Charge. |
| 5214h | 00h | RO | EBC_LSB | EBC_MSB | | | Estimated battery capacity. |

**Binned Ah (5209h)** There are ten ranges of discharge rates. Each sub-index (00–09h) holds the number of Ah discharged at that rate.

## HISTORICAL RECORDS

A historical record is generated when one of these four events occurs:

- A given number of Ampere-hours have been drawn from the battery
- A battery cycle has completed
- The battery charger has been removed
- The unit has been disconnected from the battery.

In a typical application, the Acuity will generate 15 to 20 historical records per day. The historical records are stored in the Acuity's non-volatile memory and can be read through the CAN interface.

The following three objects are used to read historical record data.

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5300h | 00..15h | RO | LSB | MSB | 0 | 0 | HIstorical record request.<br>Bytes 6+5 = record number.<br>Sub-index = index to 4 bytes within record. |
| 5301h | 00h | RO | LSB | MSB | 0 | 0 | Number of records saved; 16-bit number. |
| 5302h | 00h | RO | LSB | MSB | 0 | 0 | Bytes 6–5 battery cycle number. Returns historical record index. |

The Acuity can store up to 64,000 historical records. Each record in 64 bytes long. Record data is retrieved 4 bytes at a time. Allocation of parameter bytes within each historical record is as follows.

| SUB-INDEX | BYTE | DESCRIPTION |
|---|---|---|
| 00h | byte 0 | Runtime counter LSB (seconds). |
| 00h | byte 1 | Runtime counter LSB. |
| 00h | byte 2 | Runtime counter Byte 3. |
| 00h | byte 3 | Runtime counter MSB. |
| 01h | byte 4 | Historical Ah discharge LSB. |
| 01h | byte 5 | Historical Ah discharge Byte 2. |
| 01h | byte 6 | Historical Ah discharge MSB. |
| 01h | byte 7 | Historical Ah charge LSB. |
| 02h | byte 8 | Historical Ah charge Byte 2. |
| 02h | byte 9 | Historical Ah charge MSB. |
| 02h | byte 10 | Highest battery voltage measured during this interval (LSB). |
| 02h | byte 11 | Highest battery voltage measured during this interval (MSB). |
| 03h | byte 12 | Lowest battery voltage measured during this interval (LSB). |
| 03h | byte 13 | Lowest battery voltage measured during this interval (MSB). |
| 03h | byte 14 | Current measured at highest voltage (signed byte, units of 10 A). |
| 03h | byte 15 | Current measured at lowest voltage (signed byte, units of 10 A). |
| 04h | byte 16 | Highest temperature measured (signed byte, -40°C to 85°C). |
| 04h | byte 17 | Lowest temperature measured (signed byte, -40°C to 85°C). |
| 04h | byte 18 | Highest SoC during this interval. |
| 04h | byte 19 | Lowest SoC during this interval. |
| 06h | byte 24 | Estimated remaining Ampere Hours at the 6 hour rate. |
| 06h | byte 25 | Flags_1. *See explanation in chart below.* |
| 06h | byte 26 | Highest 1 second Avg Battery Voltage (LSB). |
| 06h | byte 27 | Highest 1 second Avg Battery Voltage (MSB). |
| 07h | byte 28 | Lowest 1 second Avg Battery Voltage (LSB). |
| 07h | byte 29 | Lowest 1 second Avg Battery Voltage (MSB). |
| 07h | byte 30 | 1 second Avg Current measured at highest Avg Voltage. |
| 07h | byte 31 | 1 second Avg Current measured at lowest Avg Voltage. |
| 08h | byte 32 | Activity_TimerD (LSB). |
| 08h | byte 33 | Activity_TimerD. |
| 08h | byte 34 | Activity_TimerD. |
| 08h | byte 35 | Activity_TimerD (MSB). |
| 09h | byte 36 | Seconds (time and date historical record was stored). |
| 09h | byte 37 | Minutes. |

| SUB-INDEX | BYTE | DESCRIPTION |
|---|---|---|
| 09h | byte 38 | Hours. |
| 09h | byte 39 | Day. |
| 10h | byte 40 | Month. |
| 10h | byte 41 | Year. |
| 10h | byte 42 | Event dependent; *see Byte 42–51 chart below.* |
| 10h | byte 43 | Event dependent; *see Byte 42–51 chart below.* |
| 11h | byte 44 | Event dependent; *see Byte 42–51 chart below.* |
| 11h | byte 45 | Event dependent; *see Byte 42–51 chart below.* |
| 11h | byte 46 | Event dependent; *see Byte 42–51 chart below.* |
| 11h | byte 47 | Event dependent; *see Byte 42–51 chart below.* |
| 12h | byte 48 | Event dependent; *see Byte 42–51 chart below.* |
| 12h | byte 49 | Event dependent; *see Byte 42–51 chart below.* |
| 12h | byte 50 | Event dependent; *see Byte 42–51 chart below.* |
| 12h | byte 51 | Event dependent; *see Byte 42–51 chart below.* |
| | byte 52–64. | [not used] |

The type of event is recorded in historical record Byte 25 Flags_1. The bit assignment for the flag is as follows.

| BYTE | BIT | ASSIGNMENT |
|---|---|---|
| byte 25 | bit 0 | 0 = lowest SoC occurred before highest SoC. |
| | bit 1 | 0 = current resolution in 10 A units. 1 = current resolution in 1 A units. |
| | bit 2 | 0 = lowest temperature occurred before highest temperature. |
| | bit 3 | 0 = lowest Avg temperature occurred before highest Avg temperature. |
| | bit 4 | 0 = lowest voltage occurred before highest voltage. |
| | bit 5 | 1 = end of charge event. |
| | bit 6 | 1 = power up event. |
| | bit 7 | 0 = battery cycle event. |

The data recorded per the events in Bytes 42 – 51 is as follows.

| | | BATTERY CYCLE EVENT | |
|---|---|---|---|
| **BYTE** | **DESCRIPTION** | **END OF CHARGE** | **POWER DOWN** |
| byte 42 | Battery cycle number (LSB). | EoC_V_LSB | PowerDown_sec |
| byte 43 | Battery cycle number (MSB). | EoC_V_MSB | _min |
| byte 44 | Total Ah charge for battery cycle (LSB). | EoC_I_LSB | _hour |
| byte 45 | Total Ah charge for battery cycle (MSB). | EoC_I_MSB | PowerDown_Day |
| byte 46 | Total Ah discharge for battery cycle (LSB). | [not used] | _Month |
| byte 47 | Total Ah discharge for battery cycle (MSB). | [not used] | _Year |
| byte 48 | Max temperature for battery cycle. | [not used] | [not used] |
| byte 49 | Min temperature for battery cycle. | [not used] | [not used] |
| byte 50 | 5Hr rate capacity available (LSB). | [not used] | [not used] |
| byte 51 | 5Hr rate capacity available (MSB). | [not used] | [not used] |

## RETRIEVING HISTORICAL RECORDS

Historical records are generated for four different events (see page 24). To find the precise historical data location where the data cycle has been recorded, use the following steps. This way data for any cycle number can be retrieved without a search through all the historical records. The retrieval starts with a request to 5302h to find the record number. (See also, chart at bottom of page 24.)

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5302h | 00h | RO | Cycle number (LSB) | Cycle number (MSB) | 0 | 0 | Read historical record. |

The Acuity response to this request is as follows.

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| 40h | 02h | 53h | 00h | Historical Record Number (LSB) | Historical Record Number (MSB) | 0 | 0 |

After the historic record number is received, specific details for that record can be retrieved, using the SDO object Historic Record Request (5300h), along with the appropriate sub-index.

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5300h | 00..15h | RO | Record number (LSB) | Record number (MSB) | 0 | 0 | Read historical record. |

The data is retrieved 4 bytes at a time. The sub-index points to a 4-byte block within the record. Sub-index 0h to 15h can be used to retrieve all 64 bytes of data.

**Example:**

The following SDO command will read historical record number 0104h. Sub-index 02h points to Bytes 8 –11 of that record.

| INDEX | SUB-INDEX | ACCESS | BYTE 5 | BYTE 6 | BYTE 7 | BYTE 8 | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| 5300h | 02h | RO | 04h | 01h | 0 | 0 | Read historical record at 0104h. |

The Acuity response to this request is as follows.

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| 40h | 00h | 53h | 02h | Historical Record Byte 8 | Historical Record Byte 9 | Historical Record Byte 10 | Historical Record Byte 11 |

# 9 — SPECIFICATIONS

The specifications for the Curtis Model 1030 Acuity are presented in Table 9-1.

Table 9-1 Specifications: 1030 Acuity

| | |
|---|---|
| **Electrical** | |
| **Operating voltage range** | 24–48 VDC; 72–144 VDC |
| **Electrical isolation** | 500 VAC, per UL 583 |
| **Transients** | IEC 6100-4-4, test level 2 |
| **Reverse voltage protection** | Acuity will not be damaged if connected to the battery with inverted polarity. |
| **Short circuit protection** | All inputs and outputs (except CANbus) shall withstand continous short circuit to B– or B+. |
| **CANbus Isolation** | Eliminates ground loops that can cause component damage as well as data errors due to differences in ground potentials among the nodes on the CANbus. |
| **Environmental** | |
| **Operating/storage temperature range** | –30°C to 55°C |
| **Humidity** | 100% condensing, per IEC 60068-2-30, Db |
| **Protection** | IP67, per EN60529 |
| **Vibration** | IEC 60068-2-6, Fc |
| **Shock** | IEC 60068-2-29, Eb |
| **Chemical resistance** | Immune to the effects of contact with battery electrolyte, hydraulic fluid, water, baking soda. |
| **EMC** | |
| **Emission** | EN55022 Class B (component test): EN12895 (vehicle test) |
| **Immunity** | EN61000-4-3 (component test): EN12895 (vehicle test) |
| **ESD** | EN61000-4-2 (component test): EN12895 (vehicle test) |
| **Regulatory Approvals** | |
| **UL (pending)** | Recognition or component listing (UL 583) |

| MODEL NUMBER | VOLTAGE RANGE | CAN TERMINATION RESISTOR 120Ω | CAN CONNECTOR |
|---|---|---|---|
| 1030-304 | 24–48 VDC | no | Bullet |
| 1030-310 | 24–48 VDC | no | Deutsch |
| 1030-306 | 24–48 VDC | yes | Deutsch |
| 1030-308 | 24–48 VDC | yes | Bullet |
| 1030-305 | 72–144 VDC | no | Bullet |
| 1030-311 | 72–144 VDC | no | Deutsch |
| 1030-307 | 72–144 VDC | yes | Deutsch |
| 1030-309 | 72–144 VDC | yes | Bullet |