# PowerDash Datalogger
# Installation and Configuration

# Table of Contents

# Logger System Basics

## *System Architecture*

Before describing the logger itself, the overall system architecture must be described, along with certain basic assumptions.

- The PowerDash datalogger is primarily an "Internet appliance". It does not use a local PC. It talks to the Internet.
- Long term data storage is in an Internet-attached database, and all user interface tools reside on the Internet.
- These include
  - A script that services the communication sessions that are initiated by the logger. This script acts as an interface between the logger and an SQL database, which contains all configuration information, as well as the stored data.
  - Scripts that provide a user interface to the configuration database.
  - Scripts that provide user access to the data.
- This system architecture can support a very large number of logging devices.
- By default, all scripts reside on the PowerDash server. As required, the system can be ported to other server systems. Server-side software is written in PHP and the database is MySQL, both of which are very popular and widely available from web hosting services.

### The PowerDash Datalogger

The PowerDash Datalogger resides in a plastic box that is about 2.5" x 4.5" x 8". It has the following connections:

- Power - connection to a small regulated 5-volt "wall wart" power supply
- Ethernet - 10/100 BaseT - network connection. Primary communications option.
- Phone - secondary communications option (modem must be installed)
- Terminal Strip
  - "VCC" - Connects to the logger's internal 5V supply
  - "0-5V" - Analog input - Usable input range is actually 0.5V to 10 V.
  - "GROUND" - Connects to the loggers signal ground
  - "+/- 250 mV" - Analog input, with range from -250 millivolts to +250 millivolts, useful for a silicon pyranometer, such as Apogee PYR
  - "COUNT 1" - an input that counts dry-contact closures to GROUND
  - "COUNT 2" - an input that counts dry-contact closures to GROUND
  - "COUNT 3" - an input that counts dry-contact closures to GROUND
  - "COUNT 4" - an input that counts dry-contact closures to GROUND
- 1-wire A - Uses the Dallas Semiconductor "1-wire" standard for transducers.
- 1-wire B - Uses the Dallas Semiconductor "1-wire" standard for transducers.
- PORT A - RS232 - Available for communicating with supported external devices.

- PORT B – RS232/RS485 (when not being used for modem).

As shipped, the logging device will, when powered on, determine what devices are attached to it, then attempt to connect to the Internet. If it has an Ethernet, this is easy. If it doesn't, then it attempts to use the phone. It will initially dial a number that is preprogrammed into the device.

Once connected to the Internet, it makes connection with the PowerDash server, at which time it reports its current configuration and acquires any new configuration that is available. It also reports any data that has been saved.

## Extension of Input Capability using the 1-Wire Ports

Various sensor devices can be attached to the logger on the 1-Wire bus

- **Temperature** - this one is relatively easy. Dallas makes a chip that attaches to the 1-wire bus. It can be soldered to a wire pair and placed in contact with the point being measured, and you're in business.

  For measuring temperature of a fluid in a solar water heater, it is possible to simply tape the transducer to the outside of a pipe (thermally coupled to the pipe with a little thermal grease), and then insulate pipe with foam insulation. A more professional way to do it is to get a "thermowell" from someplace like Omega Instruments, mount it in an appropriate fixture, and then place the thermal probe inside of it (presumably with some thermal grease to improve performance).
- **Wind Speed and Direction** - A simple wind device (which also includes a temperature probe) is available from <a href='http://www.aag.com.mx>AAG. This utilizes the 1-wire standard, and is ready to plug into the logger via a suitable cable.
- **Humidity** - AAG also makes a 1-wire humidity transducer that is based on the DS2438 and a Honeywell humidity transducer. As with the wind device, it canbe attached directly to the logger.
- Analog to Digital converters
- Counters
- Digital level sensors
- Memories

Although the devices as provided by DS are in chip form, there are a couple of sources of the devices packaged ready to use:

- AAG is a Mexican company that is providing a variety of 1-wire transducers.
- Embedded Data Systems

These sensors all have built-in serial numbers, and the 1-wire protocol includes automatic discovery of the devices.

## Other Transducers

Here are some possibilities for other transducers that might be useful.

- **Electrical Energy Measurement** - This is the primary quantity of concern for PV systems. A good way to do this is by way of a utility-style electric meter with a pulse output, this pulse output being connected to one of the counter inputs of the logger. The pulse output gives one pulse per specific unit of energy delivered (such as once per kWh). One source of these meters is Texas Meter and Device.

Other AC power transducers are available that produce a pulse output. Continental Controls sells a unit called the WattNode that works well. Similar units are available from Davidge Controls.

The pulse outputs of these devices is wired to the “COUNT N” input of the datalogger.

- **Solar radiation** - It is often useful to monitor the output of a PV system in the context of the solar radiation that is striking it. The silicon pyranometer is a useful instrument for this. We have used a relatively inexpensive (about $100) device from Apogee Instruments (model PYR). The device nominally produces 250 mV in full sunlight. The company is willing to supply a version with about 200 mV output instead, on request, which is more suitable for our logger. The output of this pyranometer would be connected to the +/- 250mV input of the logger. It comes with a calibration certificate, and the calibration constant will be programmed into the configuration database.

  As with counting, additional channels of analog input can be provided through devices on the 1-wire bus. Please consult PowerDash.

- **Flow Measurement** - In a solar thermal system, energy is delivered via the temperature change of a fluid. To measure this, it is necessary to measure flow and temperature. One good way to measure flow is via utility grade water meters, which are often provided these days with pulse initiators. One source of these that we have used is watermeters.com.

  As with the kWh meter pulse output, this pulse output can be collected by one of the "COUNT N" inputs on the logger.

## External Device Serial Port Access

The datalogger architecture provides for the communication with additional devices, such as inverters, via the serial ports. One of the serial ports is RS232, while the other is configurable as either RS232 or RS485 (half duplex).

Device drivers currently exist for a variety of different devices, including

- SMA Inverters
- Fronius Inverters
- Outback Inverters and Charge Controllers (via the Mate)
- Xantrex GT series inverters
- Davis Vantage Pro weather stations.

Contact PowerDash for information about other devices of interest.

# Installing your PowerDash Datalogger

Here is what you should have received:

- The datalogger (note: the photo shows a cable gland installed in the logger case. It is ordinarily up to the customer to drill the case in a location appropriate to the installation, and install a cable gland, if needed).
- A “wall wart” power supply. NOTE: This is a regulated 5 volt power supply. If you wish to provide your own power, your supply must provide 5 volts +/- 5%, and be capable of supplying 300 mA. An unregulated power supply, even if it says “5 volts”, is almost certain to supply a higher voltage, and will probably blow out your logger.
- A package of mounting “ears”.
- Optionally, one or two Serial port “paddle” cards.
- You may also have ordered temperature probes (not shown), and if so, these will be included in the package.

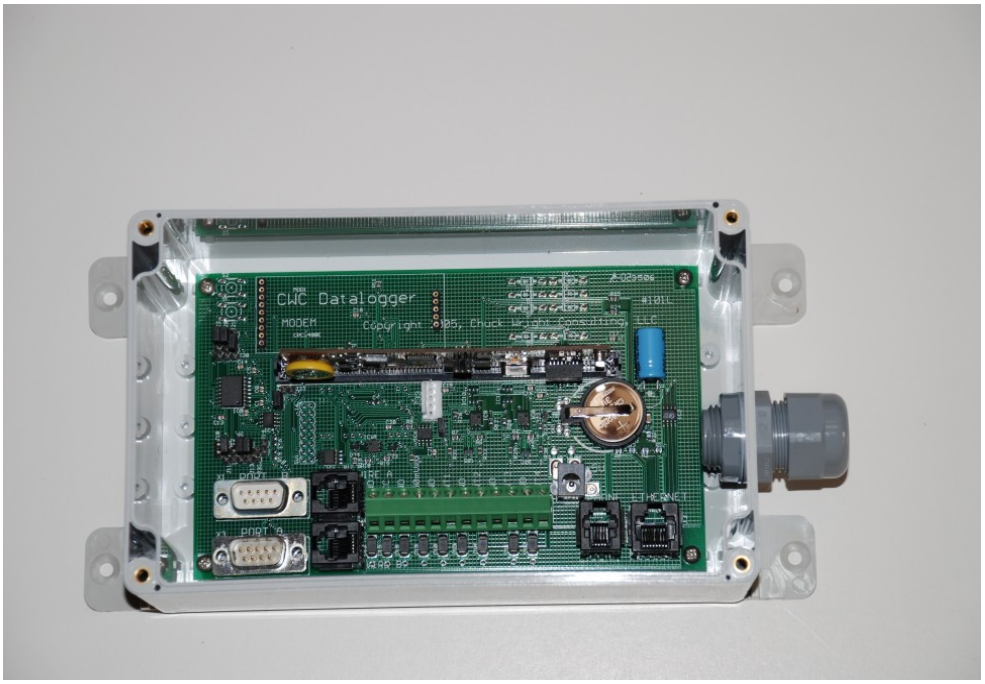
The unit was tested before shipping. As part of this test, it has already contacted our server and registered itself. If temperature probes were ordered, they have also been registered in the logger database.

Attach the mounting ears to the bottom of the box, as shown in the following photo, using the screws supplied:

## *Wiring the logger*

Remove the 4 screws attaching the cover, to reveal:

## PowerDash Datalogger




1. Mount the unit in its intended location. It will need to be within a couple of feet of an AC outlet into which the power supply will be plugged.

- Ground the unit to a solid ground using a #14 wire. The purpose of this is surge protection.Use one of the GND terminals on the datalogger.

2. Jumpers appropriate to your needs are ordinarily installed prior to shipment. You can verify this using the information in Appendix A. Jumper positions are also noted in the following section.

3. Connect devices to be monitored as follows:

- SMA Inverter (RS485 - PORT B):

- Check that the appropriate jumpers for RS485 operation are installed.
- Install the SMA RS485 daughter card in the inverter(s) according to the SMA application note.
- RS485 - wire from the RS232-RS485 converter (in the logger) to the inverters according to the SMA instructions.
  - You can use a 2-conductor plus shield.
  - Also a CAT5 works well.
- Wire inverter pins 2,7, and 5 of the inverter(s) to pins 3, 8, and 5, respectively, of a PowerDash serial adapter, and plug the serial adapter into PORT B. Inverter pins 2 and 7 should be connected using a twisted pair of approximately 120 ohm impedance. Connect Pins 5 using the shield, or an unused wire in the CAT5. A 120 ohm termination resistor should be connected

across pins 3 and 8 of the PowerDash serial adapter, and the jumper for the 120 ohm terminator in the inverter should be in place (only on the last inverter if there are multiple).

- Using the Configuration Manager, include the string "sbComm serial4" in the field labeled "Extension Control String". Multiple control strings must be separated by semicolons.

- SMA Inverter (RS232 - PORT A or PORT B):

- If PORT B is being used, install jumpers to configure it for RS232 operation. Do not install a jumper at J14 or J10.
- Install SMA daughter card in inverter, and connect to PORT A of the datalogger, according to the SMA application note.
- Using a 3-wire cable, to wire pins 2, 3, and 5 in the inverter to pins 2, 3, and 5 of a PowerDash serial adapter card, and plug the adapter card into PORT A or PORT B as required.
- Using the Configuration Manager, include the string "sbComm serial0" (if using PORT A, or "sbComm serial4" (if PORT B) in the field labeled "Extension Control String". Multiple control strings must be separated by semicolons.

- Davis Vantage Pro (and Pro2) weather station (RS232 - PORT A or PORT B):

- Install the Davis weather station according to their instructions. Install the "WeatherLink" in the console. Mount the console within a reasonable distance from the datalogger (dictated by length of supplied cable). Using the DB9 adapter that comes with the weather station, plug the WeatherLink cable into the datalogger, PORT A or PORT B.
- If PORT B is being used, install jumpers to configure it for RS232 operation. Do not install a jumper at J14 or J10.
- Using the Configuration Manager, include the string "DavisExtension serial0" (if using PORT A, or "DavisExt serial4" (if PORT B) in the field labeled "Extension Control String". Multiple control strings must be separated by semicolons.

- Davis Anemometer (direct to logger):

- Connect the RED wire (common) to any GND terminal on the logger.
- Connect the BLACK wire to one of the COUNT channel inputs. This channel will show a continuously increasing count of revolutions of the anemometer head. The logger will save a sequence of count values, along with a time (seconds) for each. To get average MPH for the interval between 2 samples, divide the difference in counts by the difference in time, and multiply by 2.25. (for m/s, use 1.006).
- Connect the YELLOW wire to the VCC terminal, through a 480 kOhm resistor.
- Connect the GREEN wire to the +/- 250 mV terminal. The voltage here is proportional to the wind direction, with 0 at North, if the support arm is facing North. Choose a calibration factor so that the max voltage (about 2.2 V) is converted to 360.
- Note: the wind direction has a “flat spot” a few degrees wide in the vicinity of North. This is inherent in the design of the instrument.

– Xantrex GT Series Inverter (RS232, PORT A or PORT B):

- If PORT B is being used, install jumpers to configure it for RS232 operation. Do not install a jumper at J14 or J10.
- Connect the Davis' RS232 cable into the desired port of the datalogger, using a PowerDash serial adapter.
- Using the Configuration Manager, include the string "XantrexGT30Ext serial0" (if using PORT A, or "XantrexGT30Ext serial4" (if PORT B) in the field labeled "Extension Control String". Multiple control strings must be separated by semicolons.

– Outback Mate (RS232, PORT A or PORT B)

- The Mate powers its RS232 drivers from the DTR (pin 4) and RTS (pin 7) signals. Using the DB9 adapter card, jumper pins 7 and 5 (GND). Connect a short wire from pin 4 on the paddle card to the VCC position on the terminal strip.
- If PORT B is being used, install jumpers to configure it for RS232 operation. Do not install a jumper at J14 or J10.
- Using the Configuration Manager, include the string "OutbackExt serial0" (if using PORT A, or "OutbackExt serial4" (if PORT B) in the field labeled "Extension Control String". Multiple control strings must be separated by semicolons.

- KYZ electric meter: If a KYZ electric meter is supplied, connect as follows:

- RED (K) - to a GND terminal on the datalogger.
- BLACK/YELLOW (YZ) - to COUNT1 and COUNT2 (or COUNT3 and COUNT4) terminals of the datalogger.

– Pulse output Water or Gas flow meters:

- Connect one wire to any “GND”
- Connect the other wire to COUNT1, COUNT2, COUNT3, or COUNT4.

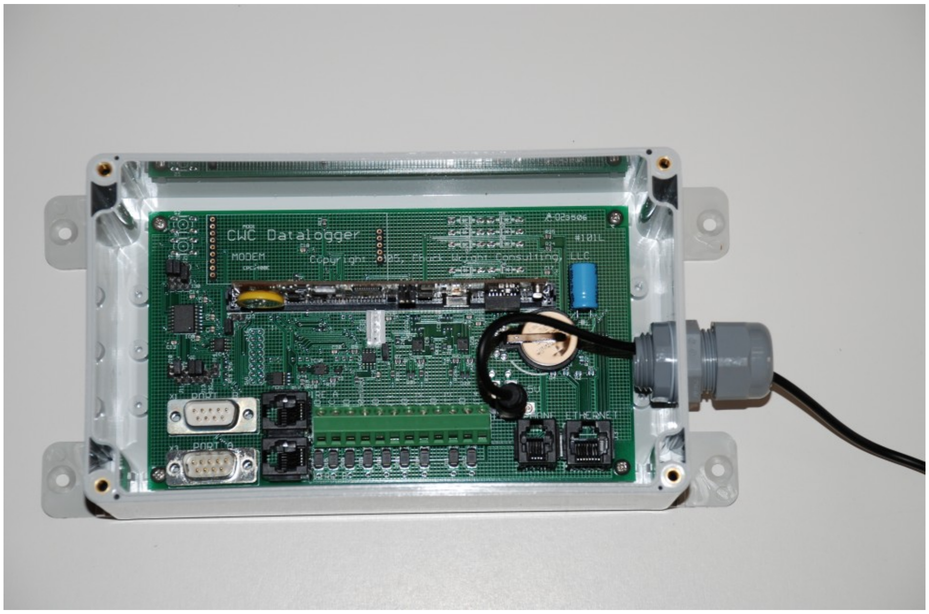
– 1-wire devices of any kind

- Plug any number in parallel into “1-WIRE A” or “1-WIRE B”. They will be detected automatically.

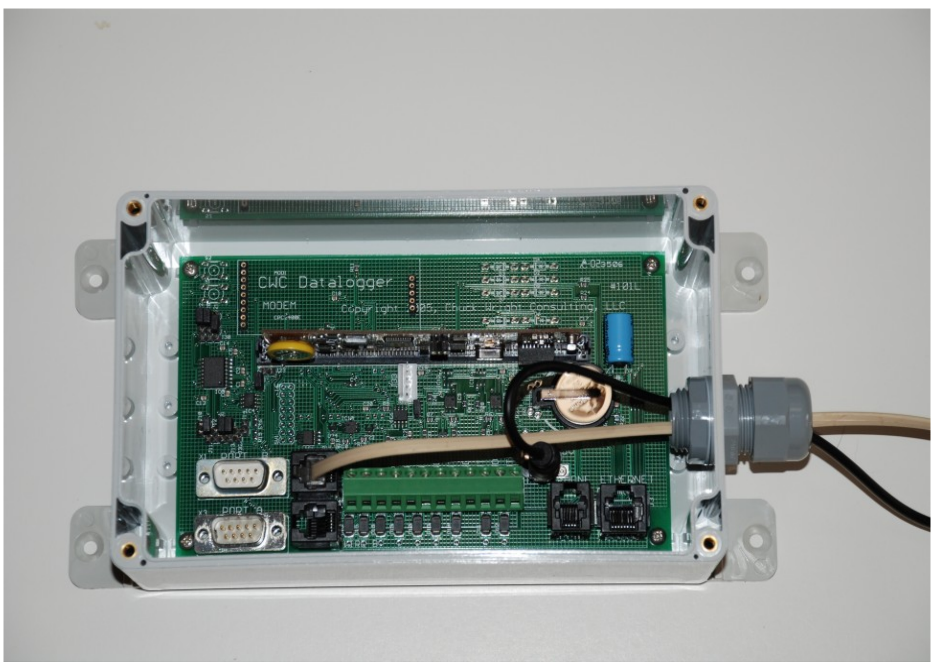
## *Wiring Hints*

Getting all required wiring through the cable gland will be easier if you are careful about the sequence.

First, run the power supply cable through the cable gland. Plug it into the appropriate connector on the logger, but don't plug in the AC wall socket yet.



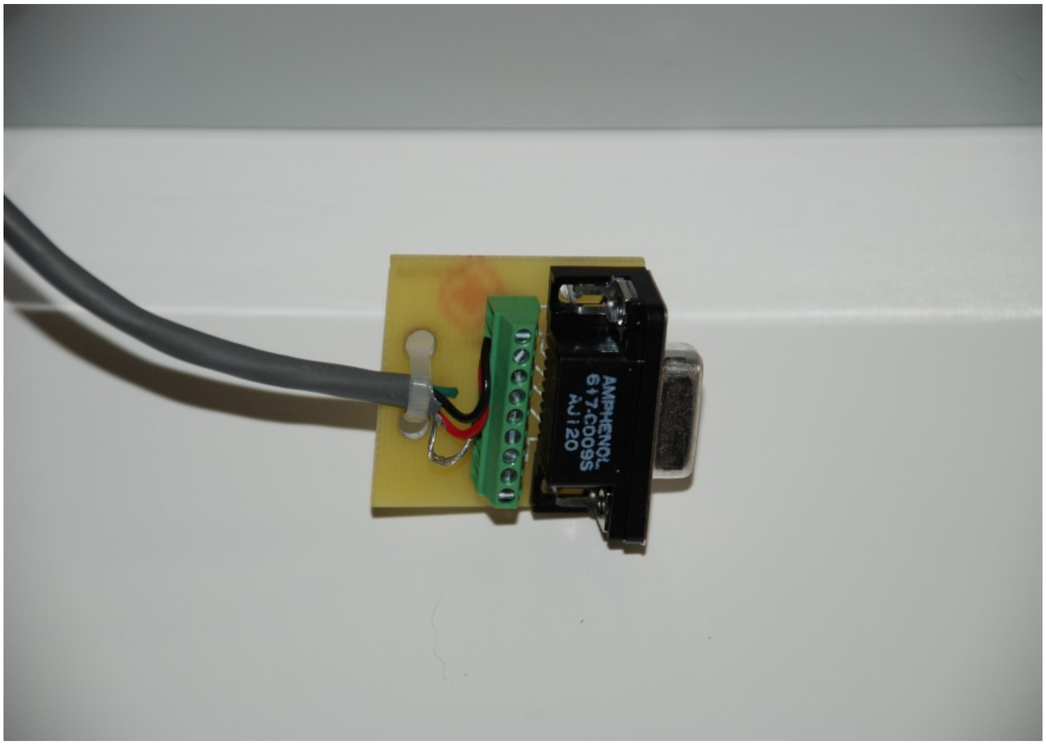

Next, run the 1-wire bus connection(s), if any. Either or both of the 1-wire sockets can be used. Discovery of 1-wire devices is automatic, and not dependent on which 1-wire port is used:

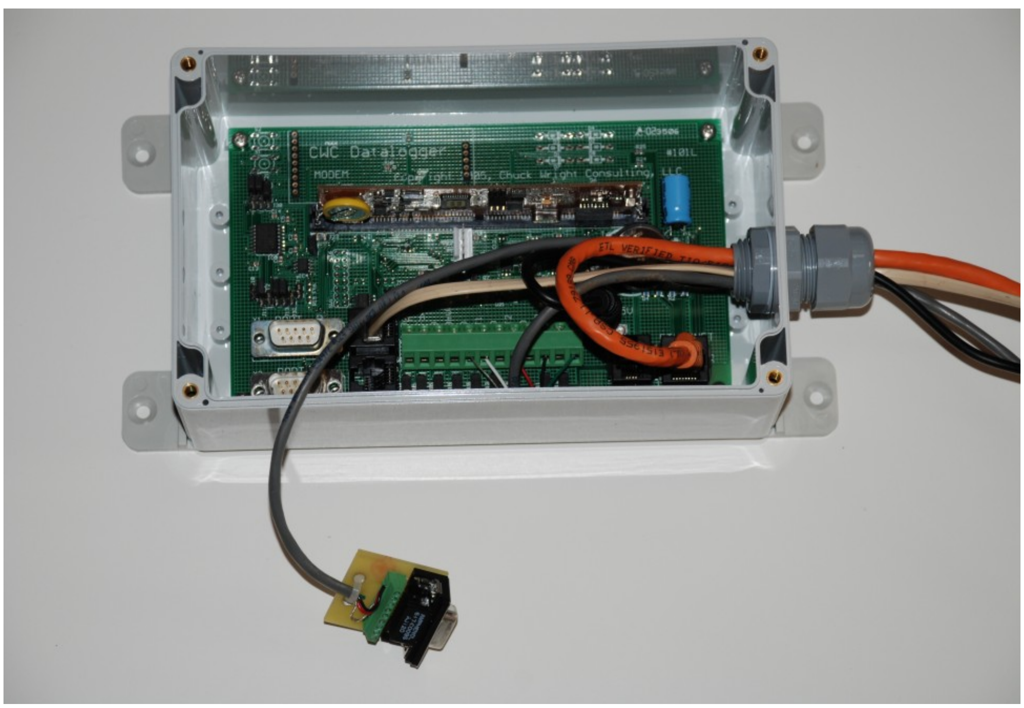



Now run the Ethernet cable (or phone line), plug it into the “Ethernet” (or “Phone”) socket. Beware that the 1-wire connections use the same kind of modular connector as a phone. Do not mix them up!:

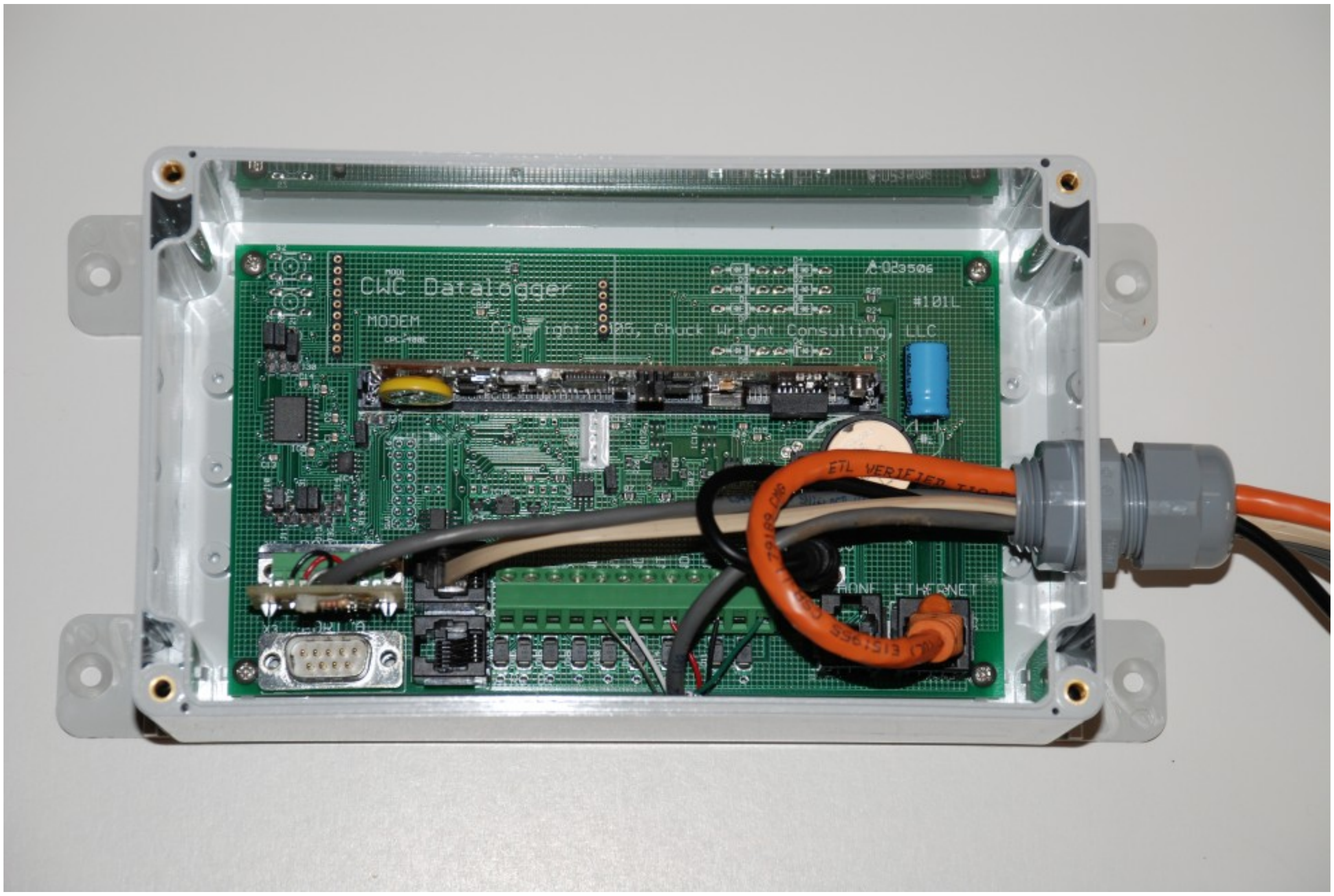

Connect terminal strip wiring, as required:


Use the paddle card(s) (if any) to wire serial ports:




And plug in to the appropriate DB9 connector:




## *Starting it up*

The most common communication configuration is Ethernet with DHCP for leasing IP addresses. If this is what you have, just plug the unit in, wait for a couple of minutes, and observe the Blinker LED codes to verify that the unit is up and running, and has communicated with the server.

If this system uses a static IP, the this was probably configured to your specs by PowerDash. If not, then **logrcfg** should be used to set and commit this configuration. See the description of this utility in the troubleshooting document.

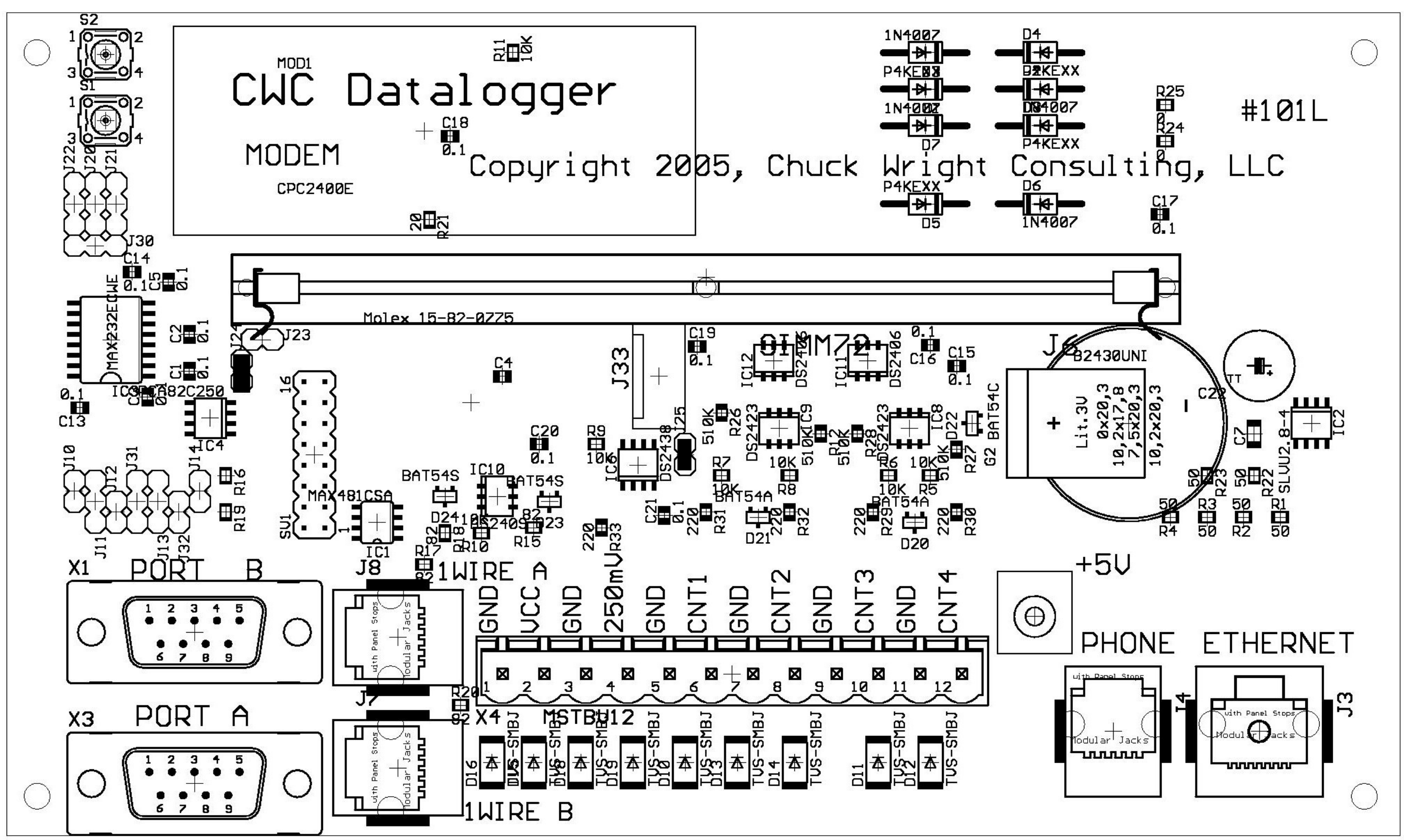
If this is a dialup system, then your specified configuration was probably configured to your spec by PowerDash. If not, **logrcfg** should be used to set and commit it.

## *Registering the Logger*

The logger has already been tested and registered with the PowerDash system. However, you will need to provide us with certain information before the logger can be used. Please visit http://www.powerdash.com/support to do this. You will need either the MAC address or the Serial Number of the unit, which are on a sticker on the processor card.

# Appendix A: Jumper Settings

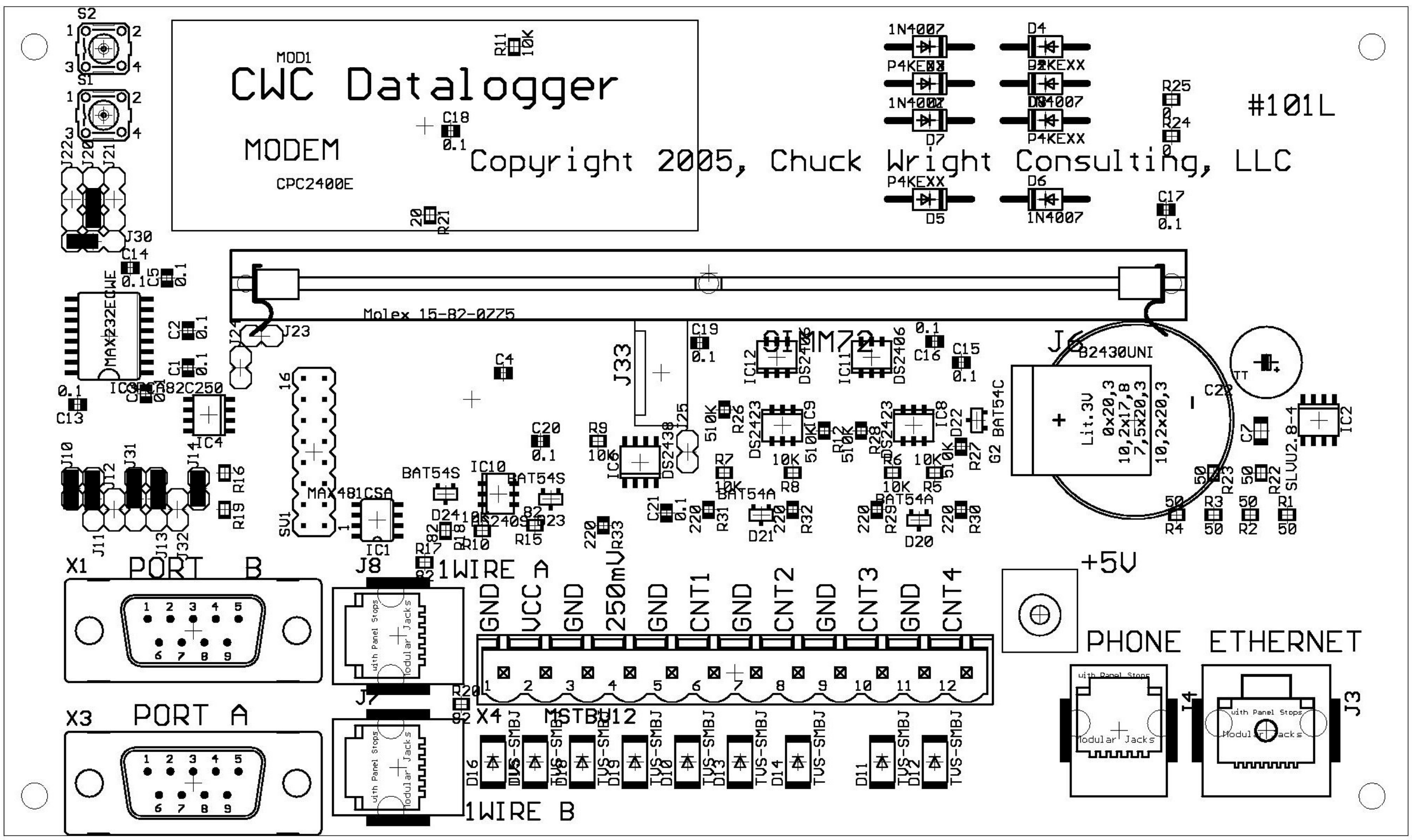
## Jumper Settings – CWC Datalogger rev 101L




### Permanent Jumpers

Jumper locations J24 and J25, marked as , are ordinarily permanently jumpered. Jumpering J24 allows the processor to respond to the reset signal issued by the JavaKit utility. J25, if jumpered, causes the firmware to immediately contact the server after a restart, instead of waiting until its next scheduled time.

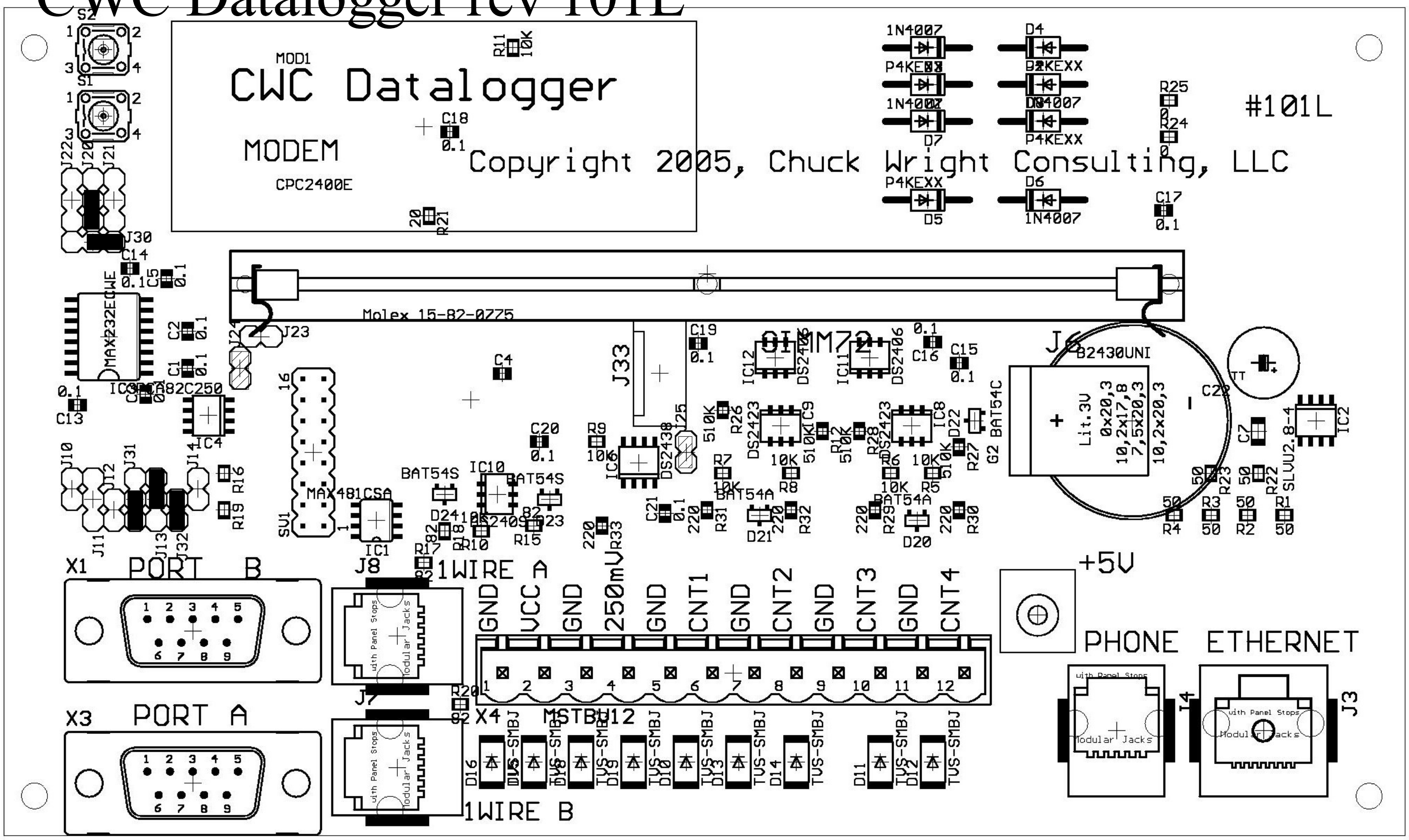
# Jumper Settings – Using PORT B for RS232
# CWC Datalogger rev 101L




### External PORT B for RS232

To use Port B for RS232, install jumpers at the locations marked as . J10 is only required if DCD is being used, and J14 only if DTR is being used (no other serial control signals are available).

# Jumper Settings – PORT B for RS485
# CWC Datalogger rev 101L



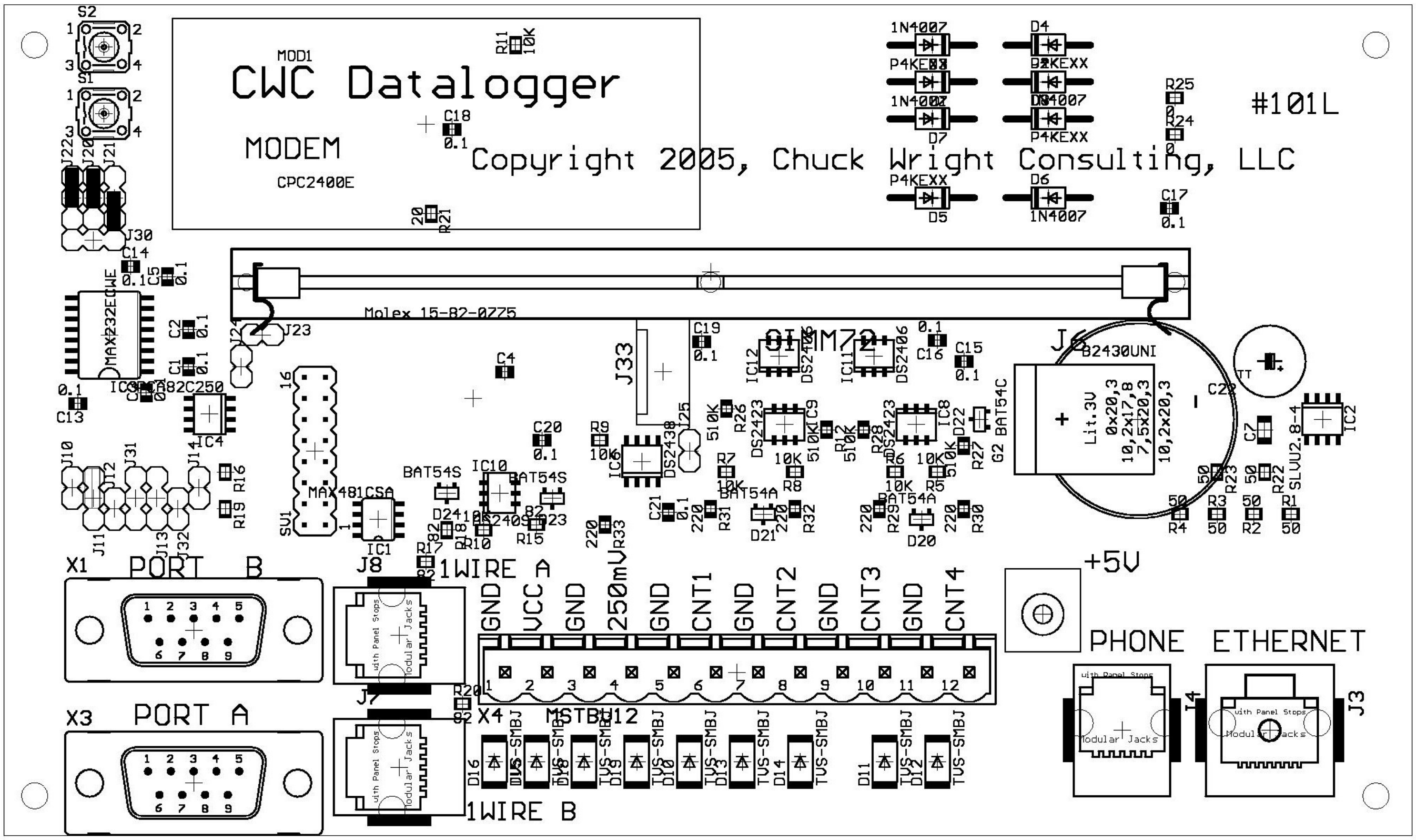

## External Port B for RS485

To use Port B for RS485, install jumpers at the locations marked as ■. These are locations J20 (lower), J13 (upper), and J32 (lower), J20 (lower), J30 (right)

RS485 wiring to Port B:

A (+) - Port A (DB9) Pin 3 – SMA Inverter Pin 2
B (-) - Port A (DB9) Pin 8 – SMA Inverter Pin 7
Common – Port A (DB9) Pin 5 – SMA Inverter Pin 5

# Jumper Settings – modem – CWC Datalogger rev 101L




## Using a modem

Jumpers should be installed in positions J20, J21, and J22 (marked ). This connects the "Serial 4" port to use the modem, instead of the external "Port B".

# Appendix B: Connecting Sensors

Transducers will be wired either directly to the logger board's terminal strip, or connected to one of the1-wire busses. This diagram shows the layout of the logger board, and the locations of the connectors.This diagram may be useful in understanding the description presented below:

## *Connecting Transducers Directly to the Logger Board*

The logger board has a screw terminal block with 8 positions,used for connecting various external transducers. The pins are:

- "VCC" - Connected to the board's regulated 5 volt power supply. Generally only used in special circumstances.
- "+/- 250 mV" - An input to an analog to digital converter. The allowed range is from negative to positive 250 millivolts(or +/- 0.25 volts), relative to the "GROUND" terminal. We have found this useful in connecting a silicon pyranometer such as is sold by Apogee Instruments. The voltage being measured should be connected between the "GROUND" terminal and this terminal.
- "GROUND" - Connected to the logger board's negative voltage.
- "COUNT 1" - Contact closures between this input and ground will increment a counter that resides in the datalogger. The counter uses the on-board battery to retain its value if power goes out. Used for connecting a variety of “pulse output” devices.
- "COUNT 2", “COUNT 3” and “COUNT 4” - Exactly like "COUNT 1".

Following are instructions for connecting some specific devices:

- kWh meter with "KYZ pulse" output (single pole dry contact closure) - wire “KY” between "GROUND" and one of the “COUNT N” inputs. Wire “KZ” to another if redundancy is desired.
- Water meter with "pulse" output (single pole dry contact closure) - wire it between "GROUND" and a "COUNT N" input.
- Apogee Pyranometer - Wire the negative wire to "GROUND" and the positive wire to "+/- 250 mV".

## *Connecting Devices to the "1-Wire" [1] Bus*

A variety of devices are available for connecting to this interface. They include wind instruments, temperature transducers, humidity transducers, switch closure counters, etc.

www.1wire.org is an excellent starting point for information about the 1-wire bus.

### 1-Wire [1] Bus Tips

This interface can be used to connect a large number of devices (networks with hundreds have been reported), over relatively long distances (wire lengths of up to 300 meters have been reported).

1 “One Wire" is a trademark of Maxim/Dallas Semiconductor.

However, to achieve reliable operation, some basic conventions should be observed.

- All devices are wired in parallel, connected anywhere, but should all be along the same basic length of wire. This means that wiring sensors like leaves of a tree should be avoided. The length of "stubs" should be kept under about 3 meters.
- For long wire runs, CAT5 cable should be used.
- Quality connectors should be used. All outdoor connections must be protected from the elements.
- If it is more convenient, use the second '1-wire' interface for some of the devices. It makes no difference which goes where.

For wiring up to a couple of hundred feet, we have had good results with standard flat, 4 conductor (only the middle 2 are used) phone cable. Many of the devices that can be bought to connect to the 1-wire bus have modular receptacles already built into them (typically 2, allowing easy chaining from one to the next). The cables can be quickly assembled using standard crimp-on modular plugs.

NOTE: Preassembled telephone cables are wired 1-4, 2-3, 3-2, 4-1. This will NOT work for the 1-wire bus. The cables must be assembled 1-1,2-2,3-3,4-4. If viewing the two ends of an assembled cable with the plugs side by side, the order of the colors will be the same (on a phone cable, they will be reversed).

Devices that already have modular sockets (such as those from aag.com.mx) can simply be chained as described above.

Some devices (temperature transducers, for example) come with bare wire leads. For these, we find that a convenient junction block is another standard telephone item, the "Dual Modular Surface Jack". This has 2 modular sockets, with the 2 sets of 4 wires wired to 4 spade lugs, screwed into the body. The "Ground" wire of the 1-wire device should be fastened to the green wires, and the "Data" wire to the red wires. If there is a third wire, labeled "Power" or something like that, it should be connected to the "Ground" wire.

Suggested color codes:

| Pin | Pin | Signal | Color (modular cable, or inside modular surface jack) | Color (Cat 5) |
|---|---|---|---|---|
| 1 | - | Vcc | - | Org* |
| 2 | 1 | - | Black | |
| 3 | 2 | Data | Red | Blu |
| 4 | 3 | Gnd | Green | Wht |
| 5 | 4 | - | Yellow | |
| 6 | - | - | - | |

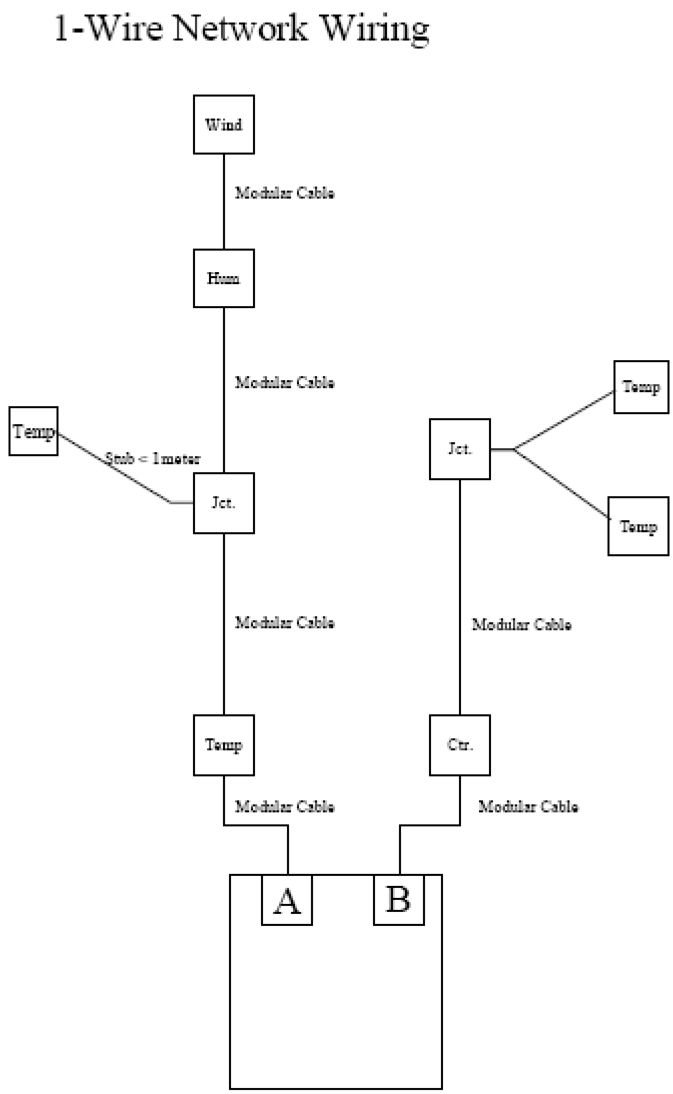
Pin Ordering (looking into socket):

```
    +------+
+---+      +---+
|              |
|              |
| 6 5 4 3 2 1 |
+-------------+
      G D   V
      N A   C
      D T   C
        A
```

(*)Ordinarily will not be Connected.

This jack can then be chained into the 1-wire bus like other1-wire device. Note that the wire from the terminal block to the device itself constitutes a stub, and should be kept under 1 meter in length.

The following diagram shows an example 1-wire network.




### *RJ11 Connectors*

We have been suggesting the use of standard telephone-style wiring components for connecting the 1-wire network. If the standard flat modular wire is used (and it is fine for relatively short distances), then modular connectors from the home center will work fine. If, however, CAT5 cable with solid conductors is being used, then it is important to use a modular connector that is made for solid wire. The following table lists some Digi-Key part numbers for suitable connectors:

| AMP PN | Digi-KeyPN | Config | Wire Size |
|---|---|---|---|
| 5-556384-3 | A9116-ND | 6-4 | 24-26 |
| 5-557965-2 | A24922-ND | 6-4 | 26-28 |
| 5-555042-3 | A9118-ND | 6-6 | 24-26 |
| 5-557970-3 | A9119-ND | 6-6 | 26-28 |
| 5-555426-3 | A9120-ND | 6-6 | 24-26 |

Using a connector that is designed for stranded wire with solid wire is likely to give unreliable connections.

# Appendix C: Logger System Operation

The datalogger device is an element in a system. Here is a simplified description of how the logger operates, and its interaction with the server(s).

The logger runs on a master cycle of one minute. Every minute, it wakes up and goes through a sequence of operations:

- When the logger is first powered on, the first thing it does is search for devices that are attached to it. It makes an internal list of these, and depending on the type of device, it decides on a default sampling schedule and policy (read on).
- If the logger is attached to an Ethernet, it recognizes this fact, and sends a request to a DHCP server to lease an IP address. If there is no Ethernet, then it makes a note to make connections via an attached modem.
- It runs through its list of attached devices, and for each, decides if a visit is necessary. It bases this decision on two parameters that are stored for each device: its sampling schedule and its sampling policy. The schedule defines a set of times at which sampling should occur, the minute(s), hour(s), day(s). The schedule specification is based on a standard that is familiar to computer people who use Unix operating systems. In that world, this method of scheduling is known as the "cron".

  The Sampling Policy defines how the logger treats the data that is acquired, and reported. We have the options of sending every sample, sending only samples that are changed from the previous value, sending an average of values taken at 1-minute intervals,and variations. As with the schedule, Sampling Policy is defined in more detail in the Functional Spec.
- As devices are visited, and their schedules and policies are evaluated, data values are selectively saved. They go into a "data queue", to await upload to the server. The data queue saves data samples, each of which consists of a time, a value, and a device id.
- Once device processing is complete, the logger consults another schedule, also in the "cron" format, and decides whether it is time to contact the server. If it is, then a separate upload thread is started.

  The upload schedule is selected as appropriate for the communication medium. If a permanent attachment to the Internet exists, then it is reasonable to connect every 5, 10, or 15 minutes, giving nearly real-time data to the server. For a dialup configuration, it may be more appropriate to program a nightly schedule, or even once every few nights.
- Once started, if there is no Ethernet, the upload thread dials an ISP and makes a connection using PPP. If this is successful, or if an Ethernet is attached, a sequence of communications with the server (or servers) is performed:
  - The current configuration is uploaded to the configuration server. This is done using the same protocol that a web browser uses to submit a form, the "HTTP POST" protocol. All of the information is placed into a string in an XML format (see Functional Spec) and the string is sent as the body of the POST.

  - When the server receives the POST, it sends the XML string to a script which parses it, and compares all of the configuration values to values in a MySQL database. In general, if any differences exist, it is assumed that the logger should be updated with the new values from the configuration database.

    The configuration script formulates a return XML string containing acknowledgment, and any updates to logger configuration that are required, and sends this as the response to the original POST operation.

    The logger parses this XML, and updates its configuration as required.

  - After the configuration has been sent and updated, it is time to send any accumulated data. As with logger configuration, the data is formed into an XML string and sent to the server. This time the data comes from the data queue, and it can be sent to the same server as the configuration, or to a different one.

    It is possible that there may be quite a bit of data in the data queue. The upload code actually sends a sequence of multiple POSTs, typically containing only a couple of thousand bytes, to the server.

    After successfully dealing with the transmitted data for a given POST, the data server returns a positive acknowledgment, again in the form of an XML string. This allows the logger to advance its data queue pointers, effectively erasing this data.

    It is worth noting that the logger is also keeping track of its own operation. In a separate data queue devoted only to recording status, it saves indicators of successful uploads, as well as any error conditions that occur in the communication process (these are more frequent than one might think, particularly with dialup connections). It sends these to the configuration server, which saves them for future problem diagnosis, and to aid in improvement of reliability.

    The data server places its data in a database, where it can be analyzed using other tools.

- Once the upload process has been completed, this thread dies. It is worth noting that the regular sampling thread carries on in parallel with the upload thread,so at times, multiple operations may be happening in the logger.

We have just described the interactions between the logger and the server-side scripts, and the databases that hold configuration and data.

There are two other parts of the system, and these both involve human interaction. The most important thing to bear in mind here is that except for installation, the human does not deal with the logger directly. The human deals with the logger through the databases that hold configuration and data.

- The first part of this is configuration management. The logger contacts its server on a schedule, and it is at this time that the logger's configuration is automatically synchronized with whatever is in the configuration database. To change the configuration of a logger, it is necessary to change the values that are stored in the configuration database. We have a script that provides a user interface to this. It reads the values from the configuration database and displays them, and gives the ability to change them as well. Any changes are made at the next scheduled upload time for the respective logger.

- The second part is the analysis and presentation of data. As with logger configuration, this function works entirely on data that is stored in an on-line database.

This system is quite a bit more complex than the typical logging system that involves simply a logger and a PC. It does not make sense to have this complexity unless some benefit is derived from it. Following are some of these benefits that we perceive:

- The logged data is available (potentially) to anyone, anywhere, who has a web browser.
- It is not dependent on operating system or type of computer.
- It does not tie up a PC.
- It opens the opportunity to monitor a large number of systems, and publicize their performance to a large audience.
- It opens the opportunity for large scale assimilation of data on the performance of renewable energy systems.
- It makes possible the monitoring of the systems by a third party. The potential exists to have free running tools analyzing the performance data and providing early feedback via email or otherwise of performance problems.
- The tool set for data analysis can easily grow, as it uses common web programming paradigms.

# Appendix D: Metering a Solar Water Heater

Measuring the performance of a solar water heater can be done with different degrees of completeness.

The easiest thing to do is to stick some temperature probes on the system, and watch the graphs of temperature. Doing this will give enough data to both learn a lot about how SWHs work, and detect when something is not working correctly. Interesting points for monitoring are

- All points at which a pipe comes in or out of the tank. The temperature probe can usually be pushed up between the insulation and the tank, and will convey a fairly good representation of the water temperature at that point.
- Pipes leading to and from a heat exchanger.
- Ambient temperature.
- Collector temperature.

The logger automatically recognizes all temperature probes that are attached to either of the 1-wire bus (power cycling is required),and they automatically appear on the configuration manager screen. If sampling is enabled, then they also automatically appear on data summary and graphing screens.

A more difficult task is monitoring the actual energy that is delivered by the SWH. To measure this, the volume of water delivered can be multiplied by the temperature change of the water.This requires 2 temperature transducers, and a flow meter. The flow meter can be placed either in the water inlet or the outlet. Flowmeters that can tolerate high temperatures cost more than those that cannot, so putting it in the cold water inlet is probably preferred.

A standard water meter with pulse output (say, one pulse per gallon)is a good way to monitor this at reasonable cost.

The temperature probes are attached to the 1-wire bus in the usual manner. The flow meter is wired to one of the logger's "counter" inputs. This device will automatically appear in the configuration manager.

One additional step must be taken: the creation of a "virtual device" to combine the temperature and flow channels into a single "water thermal energy" channel. This is done using the "Virtual Devices"screen, which allows selection of the proper devices that form the new channel.

To be completely thorough in measuring SWH performance, it is necessary to include something to monitor auxiliary energy input. This can be handled using a second counter channel, connected to either an electric meter or gas flow meter, depending on fuel source. Of course, these devices need to have a pulse output.

### *Measuring Temperature*

The temperature probes that are by far the most convenient to use with the WCS logger are those made by Maxim/Dallas Semiconductor. These connect directly to the 1-wire bus and contain unique identifiers. We use the DS18S20, which comes in a 3-legged plastic package that looks like a transistor.

We solder wires to the leads and insulate them with heat-shrink tubing. 1-wire probes can also be purchased from Embedded Data Systems, or from AAG. Of course, you can make your own.

The next question is how to effectively measure the temperature of fluid in a pipe. The "right" way is:

- Put a T fitting in a straight length of your pipe.
- Buy a "Thermowell", and screw it into the side takeoff of the T. We have found getting a good combination to be more difficult than it sounds. A combination that we have found that works is a 3/4" x 1/2" x 3/4" brass T, with a 1/2" thread thermowell with 1.25" insertion length, all bought from McMaster-Carr.
- The thermowell provides a cavity into which the thermal probe can be inserted (preferrably with some thermal grease). The large opening at the back of the thermowell should then be stuffed with insulation. Then the whole thing should be insulated.

The problem with all of this is that it is kind of expensive. The thermowells run about $18, and the T about $5. Plus the whole assembly is kind of a pain to insulate properly.

A solution that is far simpler and cheaper, and probably "good enough" for our purposes is to mount the thermal probe directly to the outside of copper pipe. Squish a dab of thermal grease between it and the pipe, and fasten the probe tightly in place with a tie-wrap. Regular pipe insulation will fit right around the whole thing. Let the wires from the probe run a couple of inches along the pipe before bring them out from under the insulation, to reduce one source of inaccuracy.

Calibration - The probes are rated at an accuracy of +/- -0.5C as shipped. It is reasonable to do at least a crude calibration to improve this. A calibration factor and offset can be entered into the configuration screen for each device. A quick thing to do is immerse the probe (protected by a plastic thermometer jacket like a nurse uses, available at drug stores) in a water/ice bath, keep stirring it, read the temperature reported by the logger, and enter an appropriate offset that will zero out the value. Or, you can fasten the probe to a reference and make the adjustment.

### *Measuring Fluid Flow*

To determine the energy that a solar water heater adds to water, it is necessary to measure the temperature of the water going into the water heater, the temperature leaving, and the flow rate. If you know these three things, then the power delivered is the specific heat of water times the temperature difference times the flow rate.

While measuring temperature is easy, measuring flow is difficult, or at least expensive, especially to get in a form that can be read by a data acquisition system. Many different technologies have been used to make flow transducers. Many are exotic and expensive.

The best thing we have found is a utility grade water meter, fitted with a reed switch to give 10 pulses per gallon. A 5/8" meter with 1/2" fittings costs about $80 at <a href='http://www.watermeters.com'>watermeters.com</a>. Accuracy is in the range of 5%. The meter provides a pair of wires, which can be connected directly between one of the "COUNTER" terminals of the logger and "GND".

The inexpensive meters are only good for COLD water. If you want to meter a hot fluid, you can pay more and get one that works up to about 200F.

# Appendix E: Extensions

The base firmware of the PowerDash datalogger supports transducers of the Maxim/Dallas Semiconductor "1-wire" variety. Other devices are accommodated within an "extensions" architecture. This architecture allows additional code modules to be activated only as they are required, and initialized in a manner that is unique to the installation.

Examples of devices that might be accommodated through the extensions architecture are

- Inverters
- Weather stations
- "virtual channels" that are implemented in the datalogger, instead of the server, which may operate on a combination of several physical devices.

Extensions are currently built into the distributed firmware module. However, they are not enabled until run time. In the future, extensions may be loaded via a dynamic loading mechanism.

Control of extensions is accomplished using the "<EXT>" XMl tag. Enclosed within this tag are one or more strings, separated by semicolons (";"). Each string represents defines how a particular extension is initialized. The first field is the name of the class that contains the extension's definition. Following that are fields that are passed to the extension's initialization routine. For example:

- sbComm serial0 - initialize class sbComm (for communicating with SMA inverters) passing it the string "serial0".
- sbComm serial0;DavisExtension serial4 - Do as above, and also initialize the class DavisExtension, passing it the string "serial4".

Even if they are built into the installed code module, extensions stay dormant until they are initialized.

As part of the upload configuration XML, the datalogger includes the <EXT> tag. The enclosed string shows all currently-activated extensions in the datalogger. It is important to note that the order in which the individual extension initialization strings appears may NOT be the same as the order in which they were originally sent by the server.

### *SMA Inverters*

The PowerDash Datalogger supports communication with the SMA ("Sunny Boy", "Windy Boy") line of grid-tied inverters via one of its serial (RS232) ports. The logger's firmware uses the SMA communication protocol to automatically discover SMA inverters that are attached to the serial port, and configures a readable channel for each of the readable inverter channels (typically about 30 channels).

The SMA protocol provides for automatic discovery of all attached devices, and for automatic discovery of the configuration of each of these devices.

To communicate with one or more SMA inverters, each inverter must be equipped with an RS232 or RS485 (if multiple inverters are present) daughter board.

Each accessible channel of the inverter appears as a single device (as would a temperature probe, for example), with its own schedule and policy. Configuration is done just as for any other sensing devices.

At present, we do not support any writing of data to the inverter.

The device driver for the SMA inverters requires the following extension control string:

```
sbComm serial<n>
```

(n is 0 for PORT A and 4 for PORT B)

### *Xantrex GT Series*

The PowerDash Datalogger supports communication with the Xantrex GT series grid-tied inverter.

This is accomplished via an RS232 interface only. Because the datalogger provides 2 serial ports, no more than 2 Xantrex GT inverters can be supported by a single datalogger.

The measurement channels that can be accessed are:

- Cumulative kWh
- Daily kWh
- Inverter Temperature
- PV Voltage
- PV Current
- PV Power
- AC Voltage
- AC Current
- AC Power
- AC Frequency
- Inverter time on
- Cumulative Wh

Use the control string

```
XantrexGT30Ext serial<n>
```

(n is 0 for PORT A and 4 for PORT B)

### *Outback FX and MX Series*

The PowerDash Datalogger has a device driver available that communicates with the Outback Mate, which is the interface box to the Outback FX inverters and MX charge controllers. This operates on either of the 2 serial ports via RS232.

A given Mate can support several inverters and charge controllers in combination. All attached inverters and charge controllers will be discovered by the logger, and a set of “devices” created.

Use the control string:

OutbackExt serial<n>

(n is 0 for PORT A and 4 for PORT B)

## *Other Extensions*

### Davis Vantage Pro Weather Station

The PowerDash Datalogger supports communication with the Davis Vantage Pro weather station via its serial port. The weather station must be equipped with the "WeatherLink" option, which provides an RS232 connection.

When the weather station is connected, and its firmware extension is enabled, each of the channels on the weather station appears as a "device" on the data logger, and has a sampling schedule and a sampling policy, just like other devices.

The firmware for accessing the Davis weather station is enabled using the extension interface, and requiring the initialization string

```
DavisExtension serialn
```

where *n* is the number of the serial port to which the weather station is attached.

Once the station has been attached and its initialization string transmitted, the respective devices will automatically appear in the configuration system.

### AAG TAI8570 Barometric Pressure Sensor

This device uses 2 separate DS2406 chips on the 1-wire bus to talk to a pressure sensor via its digital interface. The IDs of these DS2406s will appear in the device configuration list. They are also marked on the AAG device itself. These 2 device IDs should be included, in either order, as parameters in the initialization string:

```
AAGPressExt 9500000032A9E612 4100000032BE6812
```

These device IDs will always end in '12'.

After initialization, 2 new devices will appear in the device configuration list, one for barometric pressure and one for temperature. They will be named with the first of the 2 device IDs, appended with "_pr" and "_temp", respectively. For the above configuration example, the 2 additional devices will be:

```
9500000032A9E612_pr
9500000032A9E612_ta
```

The other appearances of the DS2406 chips in the device configuration list can be left unconfigured.

### AAG TAI8560 Thermocouple Sensor

This device uses a single DS2760 to read the voltage off of a thermocouple. As with the pressure sensor, the DS2760 will be automatically sensed and placed in the device configuration list. To configure the Thermocouple extension, thie device ID of the DS2760 needs to be entered in the configuration string, as in (this device ID will always end in '30'):

```
OWThermoExt 2000000000D55930 K
```

The second parameter is the type of thermocouple being used, in this type K.

After initialization, 2 additional devices will appear in the device configuration list, one with suffix of "_t", representing the thermocouple temperature, and the other with suffix of "_ta", representing ambient temperature. For the above configuration example, the 2 additional devices will be:

```
2000000000D55930_t
2000000000D55930_ta
```

## LED Signs

The datalogger supports driving simple scrolling LED signs, such as sold by Pro-Lite. Sampled data values can be substituted into text strings for display. Contact PowerDash for details.

# Appendix F: Dialup Operation

If a web-enabled Ethernet connection is not available, then the PowerDash Datalogger can utilize a dial-up connection, using the optional modem feature. When the PowerDash Datalogger is powered up, it looks for an active Ethernet connection. If this is not present, then the unit reverts to dial-up operation. **Note: Remember when testing that if an active Ethernet is sensed, the datalogger will NOT use dialup.**

On firmware versions 1.17 and later, the unit can be set to "Ethernet Only" mode. In versions 1.18p3 and later, tere is also a "Phone Only" mode. These can be set using the **logrcfg** utility. See the **Utility Programs** section for a complete description.

When a modem is used, it utilizes the "Serial 4" port, ordinarily available on the "Port B" connector. This serial port will not be available for attaching to other devices.

To utilize dial-up operation, the user needs to contract with a local ISP to provide dial-up internet access, preferably over a local phone connection. This connection may be shared with voice service. However, please consider the implications of this.

The datalogger is preprogrammed with dial-up parameters that will cause it to connect to an ISP used by PowerDash. This configuration is used after every restart, and until it is reconfigured. Ordinarily, the Configuration Manager (on the PowerDash web site) will be used to enter a set of dial-up parameters that are associated with the local ISP. In the event of a datalogger restart, the PowerDash phone number will be dialed first, at which time, the parameters of the local ISP will be reloaded into the logger.

In versions 1.17 and later, a different phone number can be committed to flash (see the **logrcfg** utility description under **Utility Programs**). This number will be used after any reset, in place of the default that is programmed into the firmware. This can only be removed by using **logrcfg** to do an "Uncommit" operation, and will persist across all types of resets.

## *Dial-up Access Configuration*

Several parameters specify dial-up operation. These revert to their default values (connecting to the PowerDash ISP) on any logger reset. The parameters should be obtained for the local ISP, and are:

- PPP Phone - Phone number for dial-up
- PPP User ID - User ID of the account
- PPP Pwd - the password to be used for the account.
- PPP DNS1 - Primary DNS for the account
- PPP DNS2 - Secondary DNS for the account

## *Things to think about*

If the dial-up service is using a shared phone line, then it is probably a good idea to program the dial-up to occur at odd times, such as at once a day at midnight. Note that data cannot be made available until after this dial-up!

Given the considerably slower data transfer rate, and the less frequent connections for data upload, the amount of data collected (number of channels and sampling period) needs to be seriously considered.

If a dedicated phone line is provided, then this is not of concern.

### IMPORTANT NOTE:

The embedded modem used in the datalogger will check that the line is unused before dialing. Also, it is set to hang up if it is on the line and someone lifts the receiver on another phone on the line. However, in such a case, the modem on the other end of the line may not hang up immediately. To make it disconnect, hang up, then lift the receiver again, and there should be a clear line.