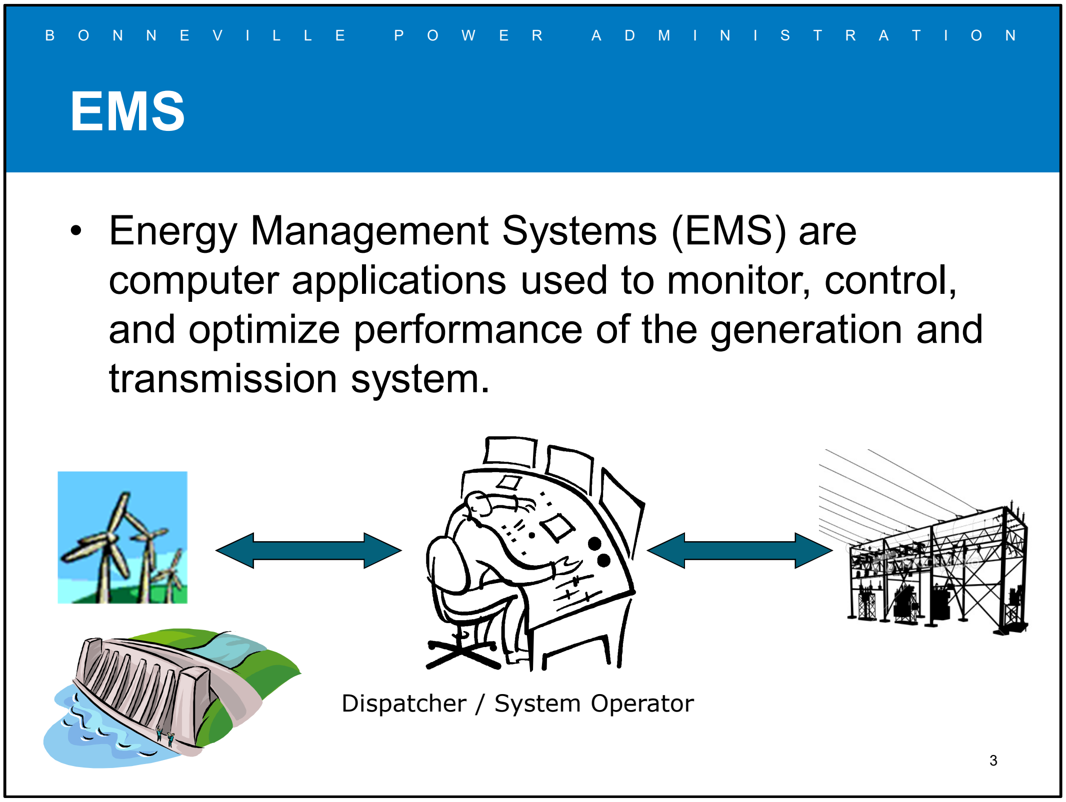
SCADA for Relay Technicians

Tracy Kealy – BPA

2019 Hands-On Relay School
Pullman, WA

Objectives

- Overview of EMS and SCADA
- Overview of DNP3
- Testing and troubleshooting DNP3 communications between IEDs and SCADA RTUs



EMS includes tools that:

- Monitor current system conditions
- Match generation to load
- Allow dispatchers to control substation equipment
- Allow dispatchers to perform “what if” scenarios
- Alert dispatchers of abnormal system conditions

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

SCADA

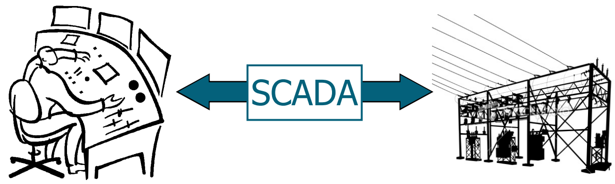
Supervisory

Control

And

Data

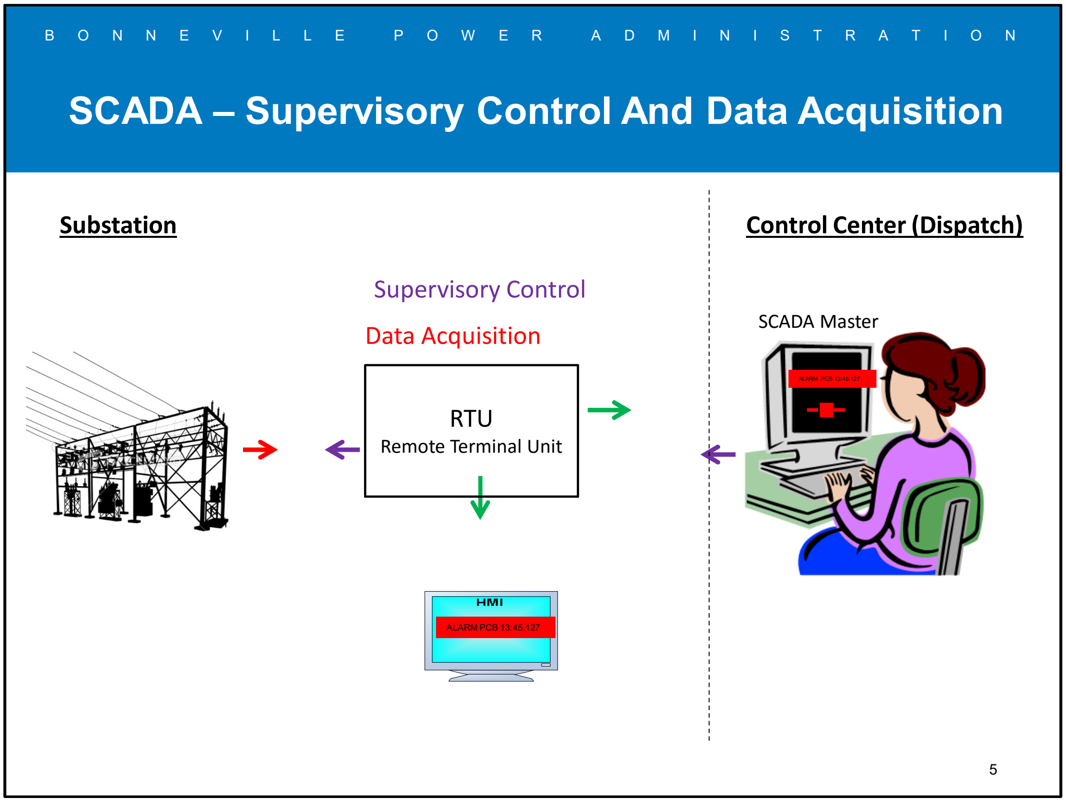
Acquisition



4

One application of EMS is SCADA.

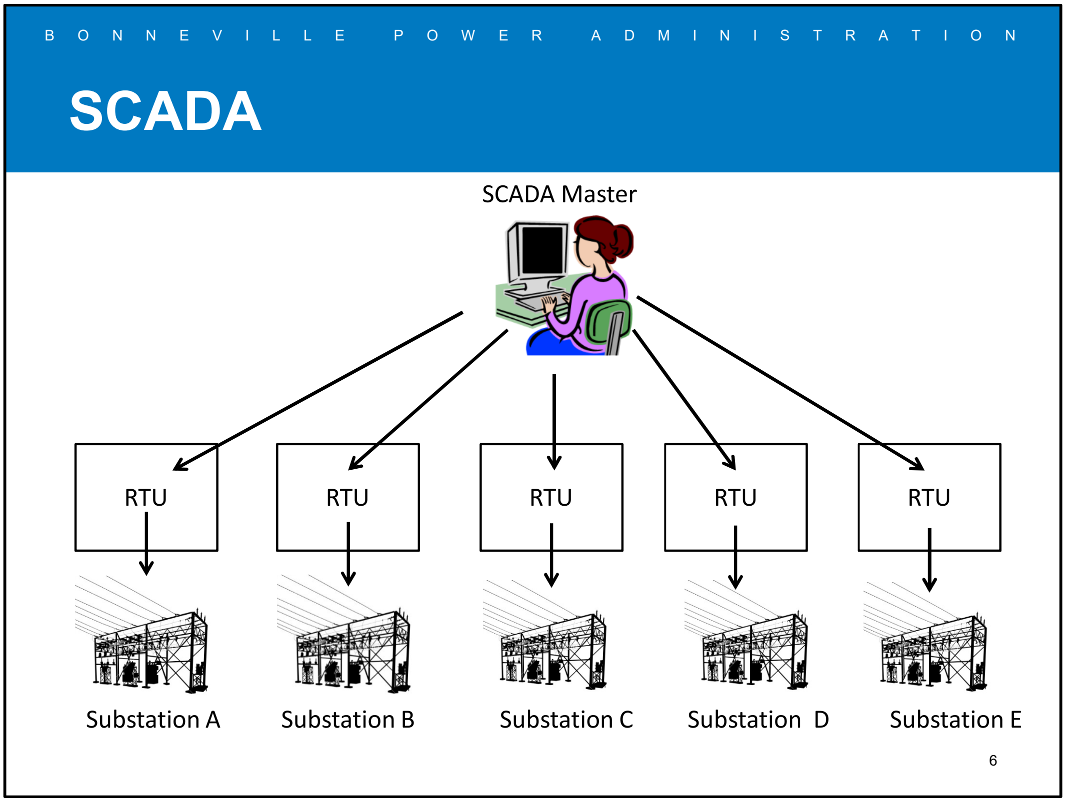
SCADA allows dispatchers to monitor current system conditions and control substation equipment remotely.



*Animated Slide.

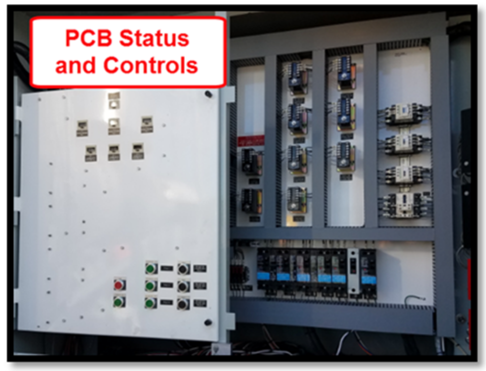
Periodically, the “SCADA master” at the control center will query or poll each of the RTUs for measurements (i.e. Bus voltage, breaker status, power flow) to gather system condition information. This is the Data Acquisition part of SCADA. Locally at the station, there may also be an HMI to provide indications at the station.

When a dispatcher wants to remotely control a piece of substation equipment (i.e. trip a breaker, raise a tap changer, etc.), they can send a command through the SCADA master to an RTU. The RTU then executes the command (i.e. close/open a contact, set an analog output value). This is the Supervisory Control part of SCADA.



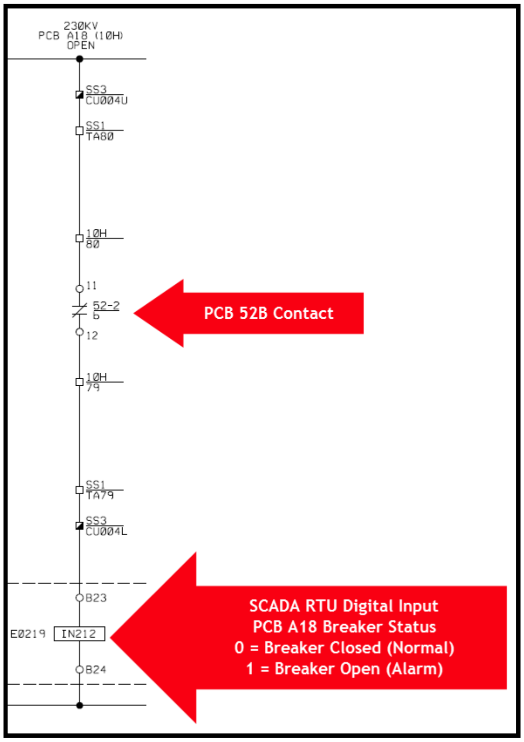
In a typical SCADA system, there is an RTU (remote terminal unit) at each substation that interface between the physical devices at a substation and a master computer at the control center.

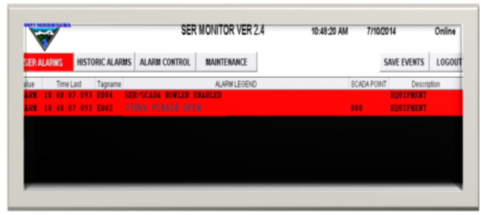
Examples of SCADA Data



An example of a SCADA data might be breaker status and control.

Example of a digital input sent to SCADA.





Alarm on local HMI and to SCADA.

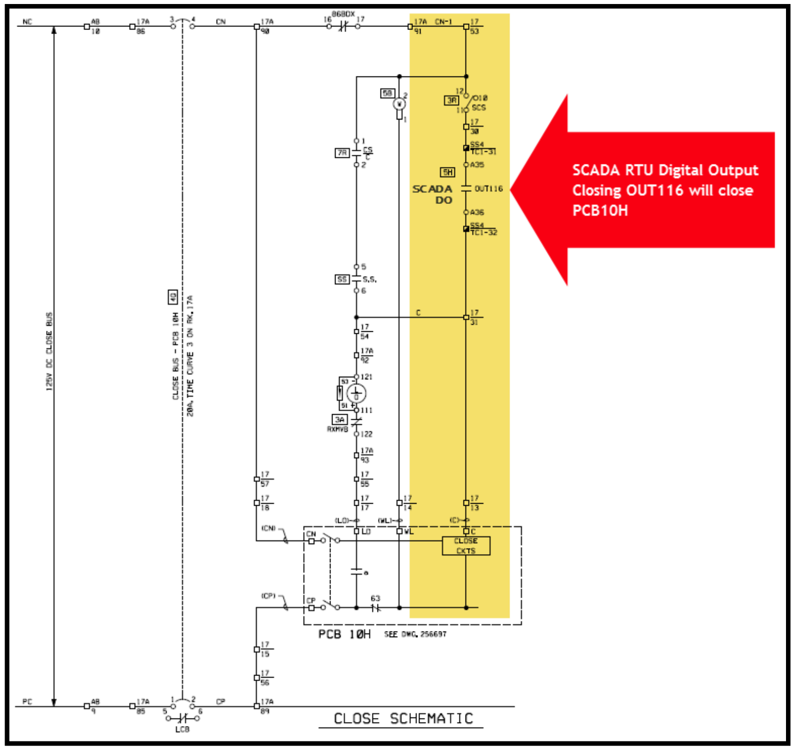
8

This is an example from an alarm schematic, which shows the 52B contact from a breaker wired to a digital input that is read by the SCADA RTU to then be sent to the SCADA master.

Binary or Digital Inputs are boolean statuses (true/false, open/close).

Digital inputs can also be used to bring in an alarm on a local substation HMI.

Example of a SCADA digital output.

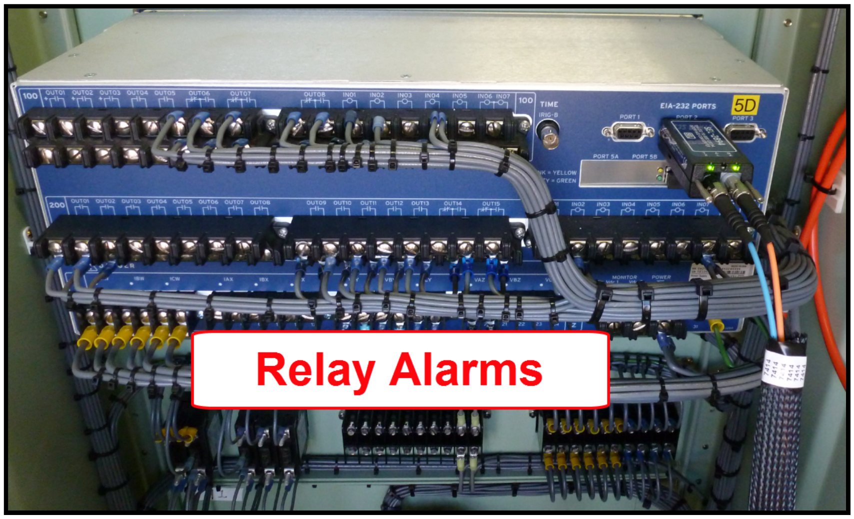


An example of a SCADA control point may be an output contact that closes in order to operate a breaker.

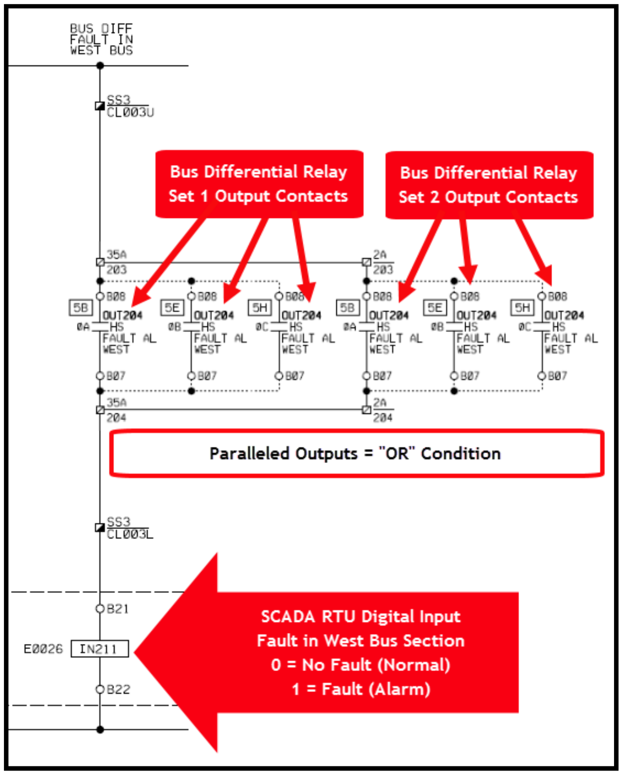
In this example, if OUT116 on this SCADA digital output module closes, the breaker will close.

Notice this circuit also has a “Supervisory Cutout Switch” which can be used to cut out SCADA control of the breaker. There can also be a local/remote switch used to put the station in local control or remote control.

Examples of SCADA Digital Inputs



Example of a digital status sent to SCADA.



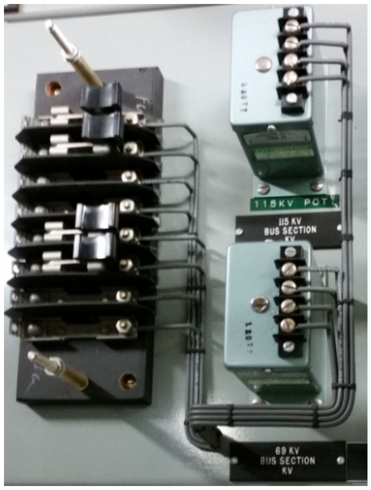
11

Here is another example of a digital status that could be sent to SCADA – this is an alarm to indicate there is a fault inside of a bus section.

For this particular alarm, the bus differential relay has a separate output contact to indicate which phase is alarming for a bus fault. Both set 1 and set 2 bus differential relay's bus fault alarms for each phase are wired in series, creating an "OR" condition (i.e. if any one of the output contacts closes, there will be an alarm).

Likewise, if alarm outputs are wired in series, this would create an "AND" condition.

Examples of SCADA Analog Inputs

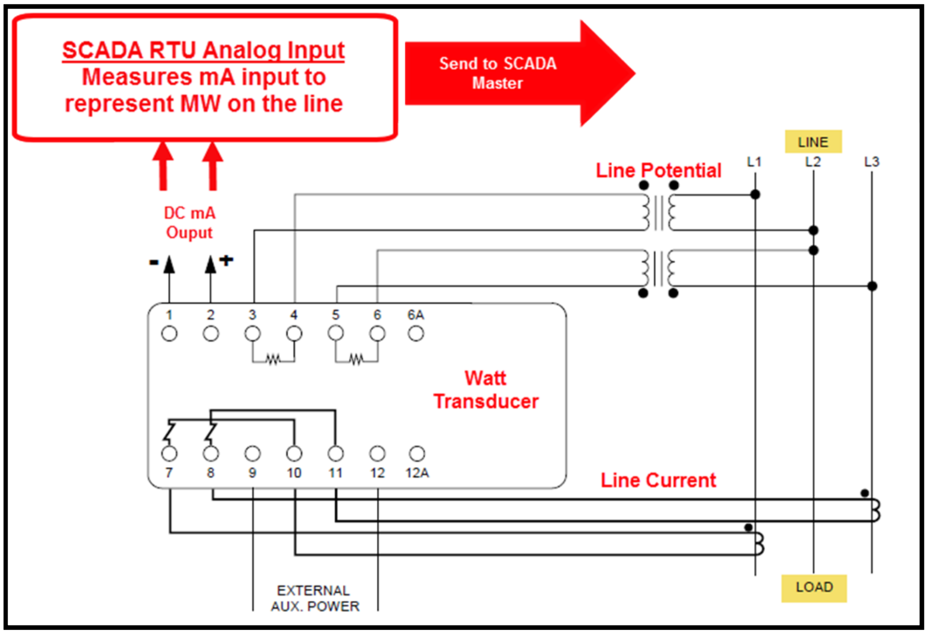


Transducers

- Voltage
- Current
- Power
- Temperature

12

Example of Analog Input to SCADA

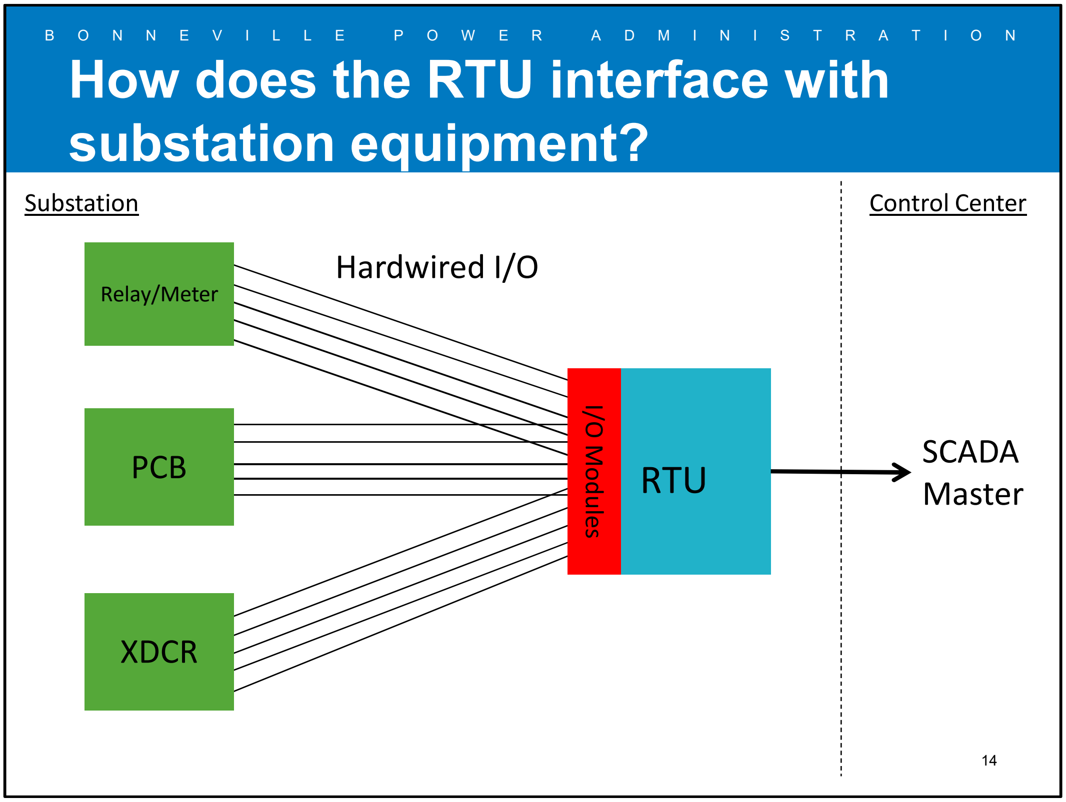


13

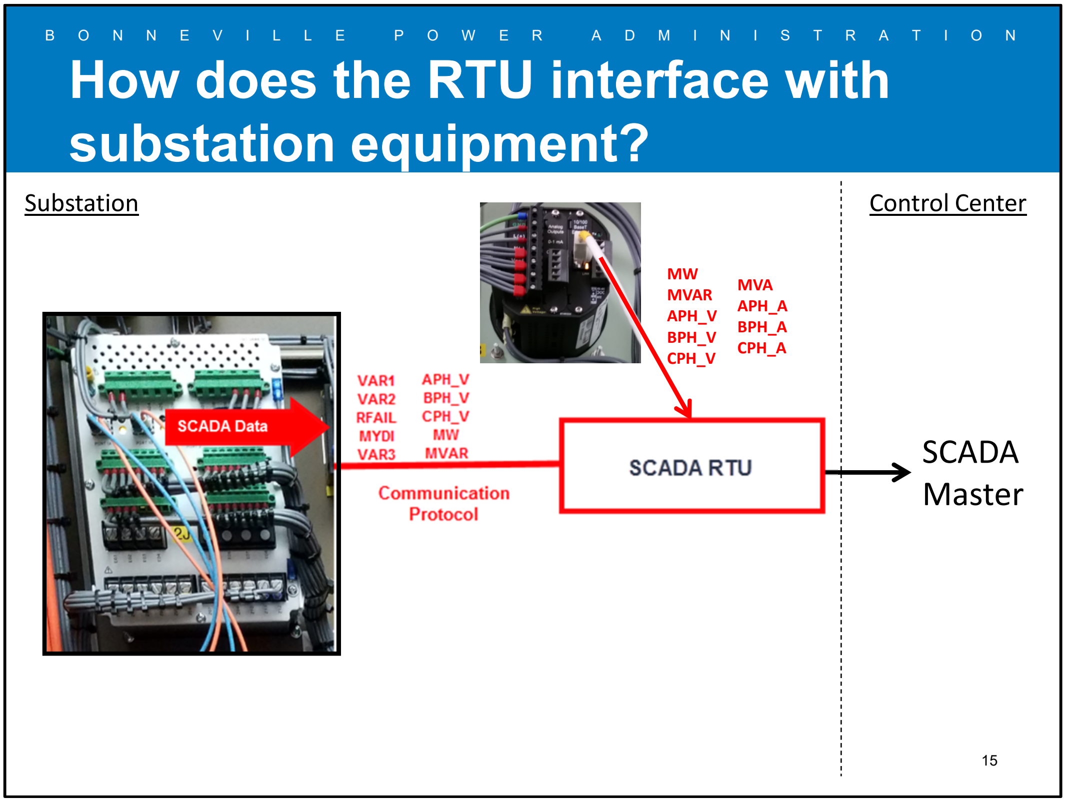
One example of an analog input that would be sent to SCADA would be power flow (MW) on a line.

In this example, a transducer is given line current and potential and produces a mA output representation of MW on the line.

This mA output is then wired to a mA input that the SCADA RTU reads and sends to the SCADA master.



As you can see, one way the RTU interfaces with substation equipment is through hardwired I/O. The RTU reads inputs and operates outputs.



Another way the SCADA RTU can interface with substation equipment is through communications protocols, which allows measurement and control data over a single serial or Ethernet cable from station IEDs (intelligent electronic devices) to the RTU. Note that communications protocols are also used between the SCADA master and the RTUs as well. This architecture reduces wiring between the IEDs and the RTU, but increases complexity in that both the IED and RTU now have to be configured to communicate using a communications protocol.

Because this class is focused on SCADA for Relay Technicians, we're going to focus on the communications between the station IEDs and the RTU. Specifically, we will be focusing on DNP3, how to configure station IEDs to communicate to a DNP3 master, and how to test and troubleshoot DNP3 communications.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

What is a Protocol?

- "A protocol is a series of prescribed steps to be taken, usually in order to allow for the coordinated action of multiple parties."
- A communications protocol is the set of rules for data representation, authentication and error detection required to send information over a communications channel.
- The protocol is like the language that a device speaks. Devices must speak the same language in order to communicate.

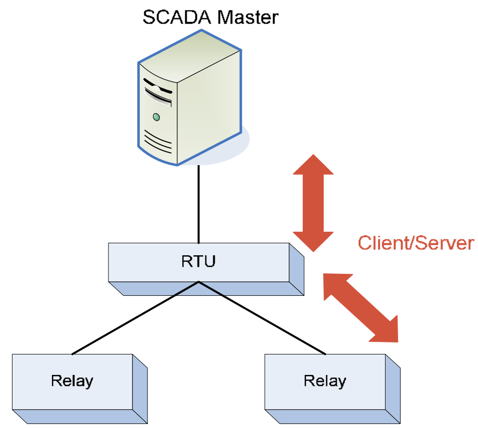
Source: http://ascherconsulting.com/what/is/the/difference/between/a/protocol/and/a/standard/

There are different definitions of a protocol, but essentially it is a set of rules that define a language so that two devices can communicate.

If you think of a language, there are rules that need to be followed and words that need to be understood with the same definition between the two parties communicating in order for communication to happen. The same is true for two devices communicating.

Types of Protocols

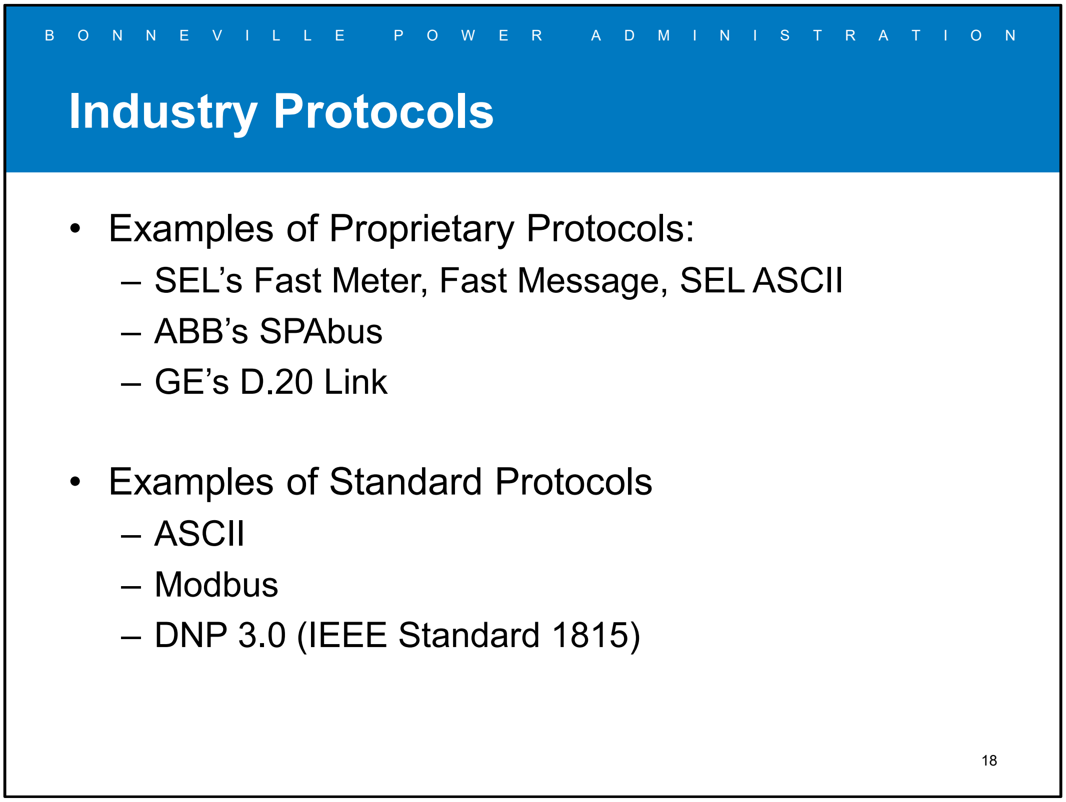
- Point to Point Protocol (PPP)
  - Establishes a direct connection between two devices.
- Client/Server or Master/Slave
  - The Master or Client initiates communications, and the Slave or Server responds and sends data.



There are different types of protocols. Point to point protocols establish a direct connection between 2 devices. Master/slave or client/server protocols have communications initiated by a master device and the slave responds.

Notice in the diagram that you can have a device that is both a master and a slave (i.e. an RTU is the master to slave IEDs in a substation, but it is also a slave to the SCADA master).

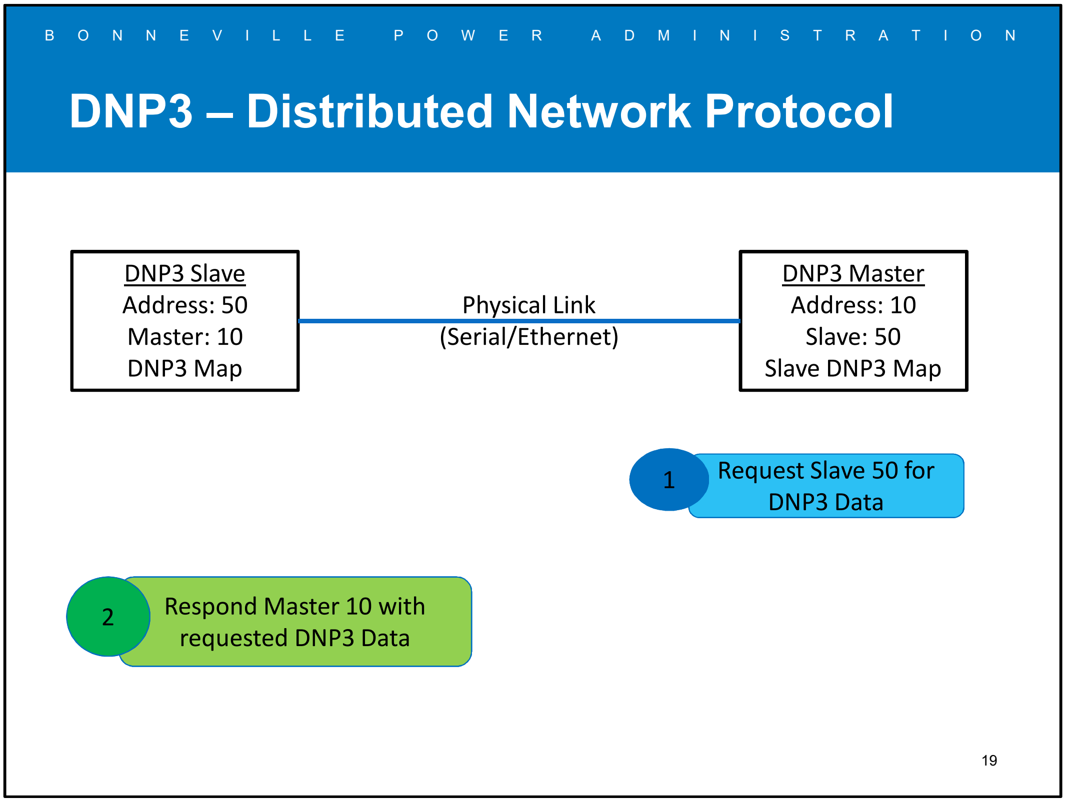
Not shown in this slide are also peer to peer protocols, which are protocols where either device can initiate communications.



There are many different communications protocols used in the industry today. Some are protocols that are manufacturer specific and some are open protocols (i.e. not specific to a particular manufacturer).

<u>Examples of Open Protocols</u>

- ASCII
  - Developed in 1963 for character coding text between computers and devices.
- Modbus
  - Created by Modicom in 1979 to be used to establish communications between their PLCs.
  - Modicom published the Modbus protocol as a free, open standard and it was quickly adopted by other manufacturers and is widely used by IEDs today.
- DNP 3.0
  - Developed by Westronic Industries between 1992-1994.
  - Westronic was an RTU manufacturer and system integrator at the time and were constantly converting the hundreds of proprietary utility protocols in use at the time to work with their devices.
  - Long story short, DNP3 was developed. The developers incorporated the best features of utility protocols in use at the time, while also using as little bandwidth as possible and making DNP3 as reliable as possible.
  - DNP3 is a widely used industry standard protocol used for SCADA applications.



**Animated Slide.

Due to the amount of time we have for this presentation, I'm only going to cover DNP3. However, note that Modbus is very similar.

DNP3 (Distributed Network Protocol) is one type of standardized communication protocol commonly used for SCADA applications to collect local substation data from IEDs.

It is a master/slave or client/server protocol (in the standard, the terms Master/Outstation are used).

The master initiates communications and sends requests for data or control commands to the slave. The slave then responds.

You might be wondering what makes a device a DNP3 master or a slave. Not all devices that need the ability to communicate via DNP3 require every feature of DNP3. In order to allow developers to only implement the parts of DNP3 that a device needs, the DNP3 standard outlines implementation levels that define which functions or features of the DNP3 standard are required for each level. DNP3 level 1 devices only require some basic functions of DNP3 and the rest are optional, whereas level 2 and 3 devices require more functions of DNP3.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

Basic DNP3 Setup

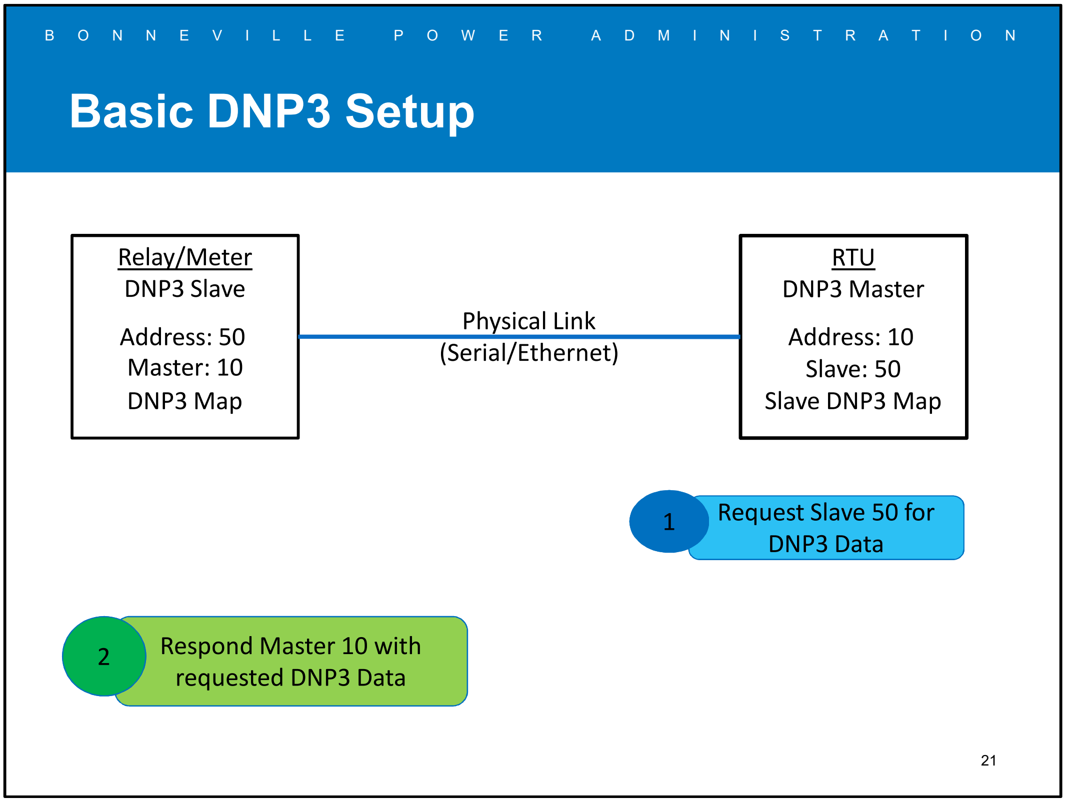
1) Physical Link
2) Addressing
3) DNP3 Maps
4) Reporting/Polling Strategy

20

When it comes to configuring two devices to communicate via DNP3, I like to break down the setup into 4 major parts.

1) Physical Link – how are the device's physically connected to establish a communications channel?
2) Addressing – who is communicating?
3) DNP3 Maps – what is the data being exchanged between the slave and the master?
4) Polling Strategy – how is the data exchanged between the slave and the master?

I don't want to gloss over the fact that when you're establishing communications between two devices from different vendors, a major challenge is dealing with the fact that the devices are configured with different pieces of software and the settings are named differently. For the most part, it should be straight forward to identify settings in these categories, but it's not always the case.



*Animated Slide

Example: Let's say you want to gather data from a relay or meter (i.e. breaker status, relay element pick up, analog quantities, kWh, etc.) using DNP3. The RTU would be the "Master" and the IEDs (relays, meters, PLCs, etc.) would be the "Slaves".

Basic Setup

1. Starting with your relay/meter, it first needs to establish a physical link to the RTU.
    - For serial links, you may need to consider:
        - Which serial port do you want to use on each device?
        - Cabling (RS-232, RS-485, manufacturer specific cabling, etc.)
            - DTE (data terminal equipment) vs. DCE (data circuit-terminating equipment)
        - Baud rate – speed in bits per second
        - Data Bits – define how many bits are used in a single data transmission.
        - Parity Bit – used for error detection
        - Stop bit – used to indicate the end of a data transmission
        - Handshaking / Flow Control – used to establish communication paths before data can be sent to avoid buffer overflow issues.
    - For Ethernet links, you may need to consider:
        - Cabling (i.e. copper or fiber)
        - IP address
        - Subnet Mask

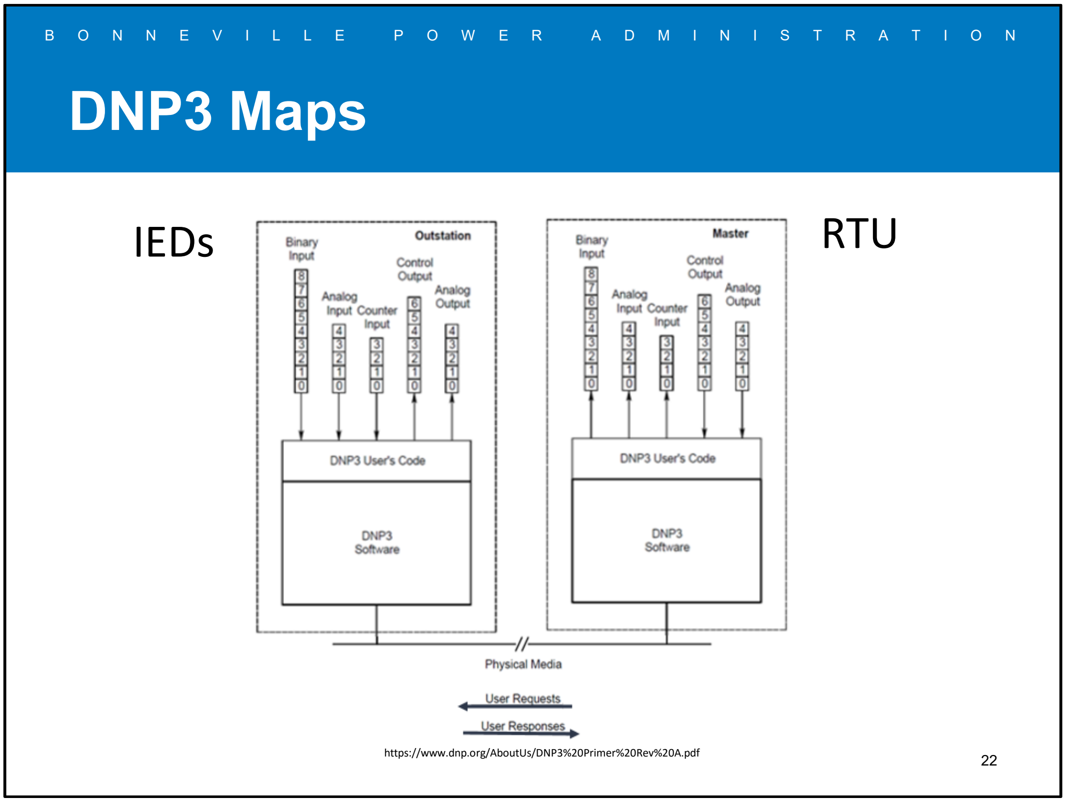
- Default Gateway
- Speed/Duplex
- Ethernet switch settings

2. Next, both the master and slave devices then need to be assigned a unique DNP3 address to they can identify each other and themselves when they communicate. There are 65520 addresses available in DNP3.

3. The slave device has a data map called the “DNP3 map” that defines the DNP3 data available for the master to query or send controls to. The master device needs to know the slave’s DNP3 map so that it can populate its database with the data from the slave device, as well as send control commands to specific points.

4. Finally, you need to decide what your reporting/polling strategy is – that is, how is the data supposed to get from the slave to the master and how should the data be formatted? Do you want the master to poll certain for certain data more frequently than others? Do you want the slave to send data unsolicited (i.e. without being polled)? Based on what your polling strategy is, the RTU needs to be configured to request certain data from the IED and the IED needs to be configured to respond to those requests in the correct format.

You may or may not be the person who determines what the DNP3 maps are or what the polling strategy should be, but as someone who is testing the relay/meter, it’s important to have a basic understanding of all of these parameters so that when you are testing/troubleshooting, you know what to expect when it comes to verifying DNP3 communications.



Let's take a look at the DNP3 data itself. In DNP3, inputs and outputs are always referenced from the master's perspective.

- Each slave device has a DNP3 database, separated by data type, which is often called a DNP3 map. The map represents the DNP3 data available in the slave for the master to poll or control.
- Each little block in this diagram represents a data point or in DNP3 terms, an "object". So, when you have multiple instances of the same data type, you can think of that data being arranged into a list or a map where each element of the list is called an "index".
    - A map of binary inputs represent boolean states (i.e. breaker status, disconnect status, relay fail alarm, etc.)
    - A map of analog inputs represent analog quantities that are measured by the outstation (i.e. ambient temperature, fault location, MW, etc)
    - A map of counter inputs represent count values, like kWH, that are incrementing over time.
    - A map of control outputs represent boolean states that can be set to on/off, raise/lower, open/close.
    - A map of analog outputs represent analog quantities that can be set to a particular value.
- One of the goals of the master device is to keep its database up to date by polling the slave or "outstation" device(s) for inputs. It can also issue commands or outputs to the outstation. The master device needs to be configured such that the requests made by the master is formatted or encoded in a way the slave device can understand and decode. The way the master polls or requests data from/to the slave devices depends on the user's polling strategy.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

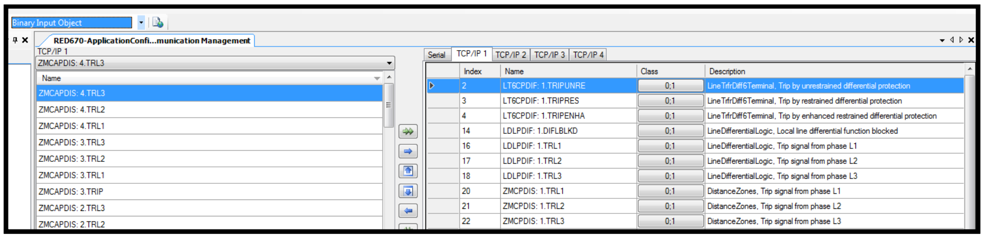
DNP3 Map Examples

Fixed DNP3 Map from a Device Manual

Table 3. Binary Input Points

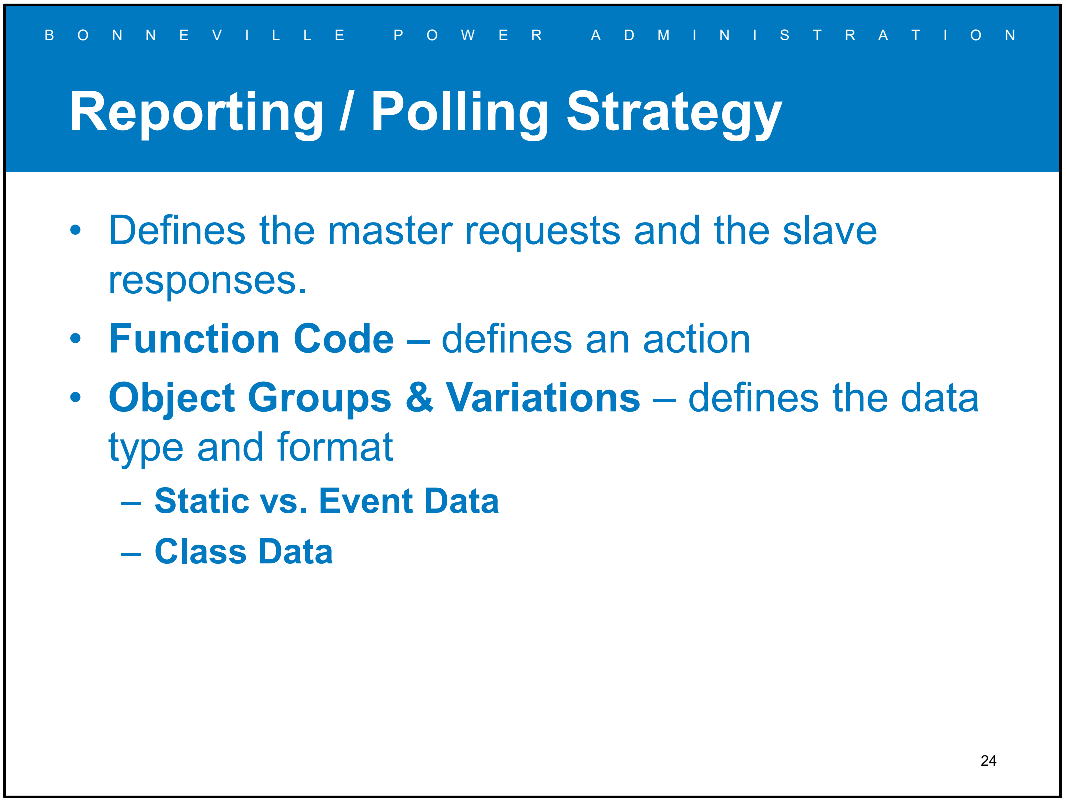
Binary Input Points	
Point Index	Description
0	50TA Phase Tripped
1	50TB Phase Tripped
2	50TC Phase Tripped
3	150TA Phase Tripped
4	150TB Phase Tripped

Custom DNP3 Map in Device Software



23

Some devices have fixed DNP3 maps and some allow you to create custom DNP3 maps.



The DNP3 maps define the data within a slave device, but the polling/reporting strategy defines how the master sends requests and how the slave responds. **It is really important to understand what the polling/reporting strategy is so that when you test and troubleshoot DNP3 communications, you know what to expect.**

Function codes define the action that needs to be taken. For example, the DNP3 master may request to read, write, or operate something and the slave device needs to respond to that request or send an unsolicited response meaning it doesn't need to be queried or polled in order to send data.

Object Groups define the data type so that DNP3 data can be encoded and de-coded in a standardized way. Object groups can also be used to develop a reporting/polling strategy.
Variation specifies how the data within an object group is encoded (i.e. data format).

Not only do you have your 5 basic data types (i.e. binary inputs, binary outputs, analog inputs, analog outputs, and counters), but other "data types" that are defined by object groups are:

- Static Data = current value
- Event Data = something significant happening (i.e. state change or analog exceeding a dead band)
- Class Data allows multiple data types to be prioritized into 4 data classes.
    - Static Class 0 Data = All Static Data that has been assigned to one of the four classes.
    - Event Class 1,2,3 Data = Slaves can attribute its event data to one of these 3 classes.

Example of a polling/reporting strategy used with class data: you can set the RTU to send an event class poll periodically to gather all event data with a single request. Then less frequently send a Class 0 poll just to make sure the RTU database is in fact up to date with the latest static values.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

Examples of DNP3 Objects

Object Group	Object Name	# of Variations
1	Binary Inputs	2
2	Binary Input Events	3
10	Binary Output	2
20	Counters	8
21	Frozen Counters	12
30	Analog Input	6
32	Analog Input Event	8
41	Analog Output	4
60	Class Objects	4

25

When it comes to reporting/polling strategy, it's important to define how the data should be formatted to meet the application's needs. There are a number of different types of data defined in the DNP3 standard. Refer to IEEE Standard 1815 Annex A - DNP3 Data Object Library for the full list.

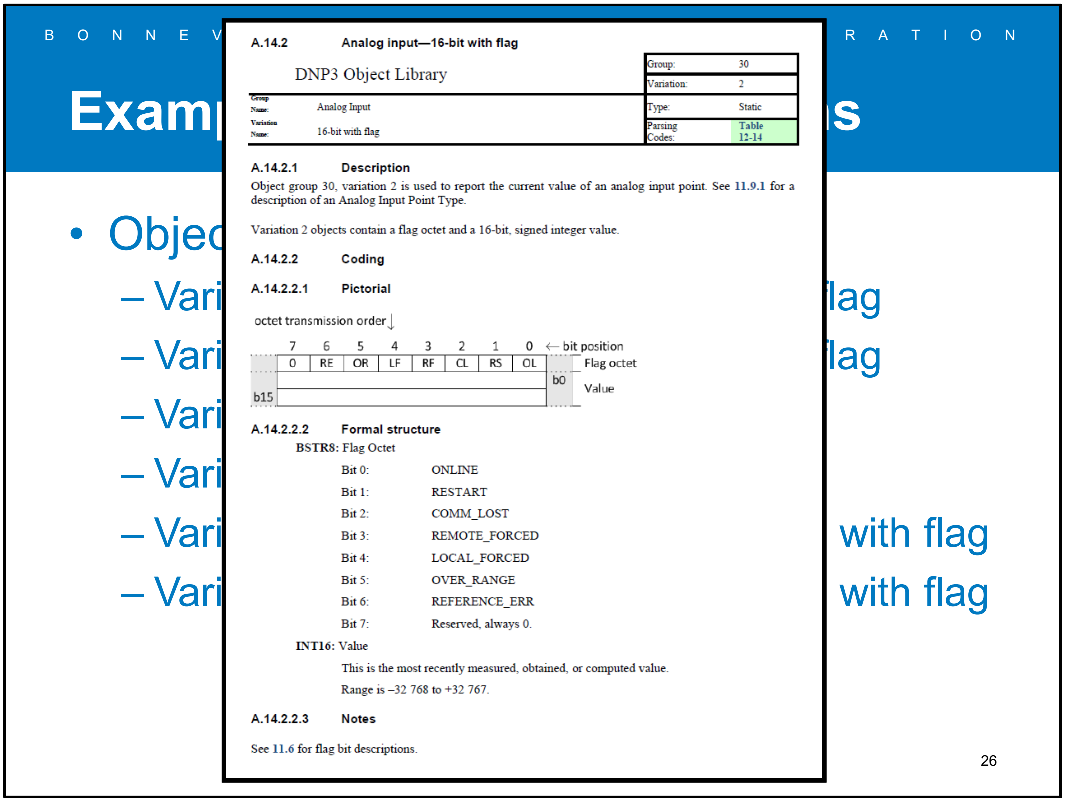
These are just a few of the DNP3 object groups defined in the standard.

You'll also notice some object groups are "events". In DNP3, there is static data that represents the present value of a point and you have event data that represents something significant like a state change of a binary input or an analog input exceeding a threshold.

Along with data types being defined by Object Groups, each Object Group also has multiple variations that define the different data formats of an object group. For example, analog inputs (Object 30) can be represented as a 16-bit integer, a 32-bit integer, or a floating point value and the variation defines that.

Earlier we mentioned how different devices have different implementation levels of DNP3 – this means that **not all DNP3 devices can transmit or interpret all DNP3 objects and variations. Also not all DNP3 devices can support all function codes available.**

It is also up to the manufacturer to decide which object groups and variations to use to represent a device's data with DNP3. So you might have one device that can represent 3-phase power in any of the Object 30 variations, but maybe another device of a different manufacturer can only support a few of the Object 30 variations.



*Animated Slide.

As an example, here are the six variations of Object 30 (Analog Inputs).

Flags are an octet of data attributes like point online/offline status, forced status, over range status, etc. that give more detail about the point rather than just the value.

Integers are whole numbers whereas floating points have decimals. You can see how depending on which variation is available for use, you may need to perform some scaling of an analog quantity in order to achieve the granularity required for a particular analog value.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

Example of DNP3 Map

Object 30 - Analog Inputs (Secondary Readings) - Read via Class 0 or with qualifier 0, 1, 2, or 6

Object	Point	Var	Description	Format	Range	Multiplier	Units	Comments
30	0	4	Meter Health	sint16	0 or 1	N/A	None	0 = OK
30	1	4	Volts A-N	sint16	0 to 32767	(150 / 32768)	V	Values above 150V secondary read 32767.
30	2	4	Volts B-N	sint16	0 to 32767	(150 / 32768)	V	
30	3	4	Volts C-N	sint16	0 to 32767	(150 / 32768)	V	

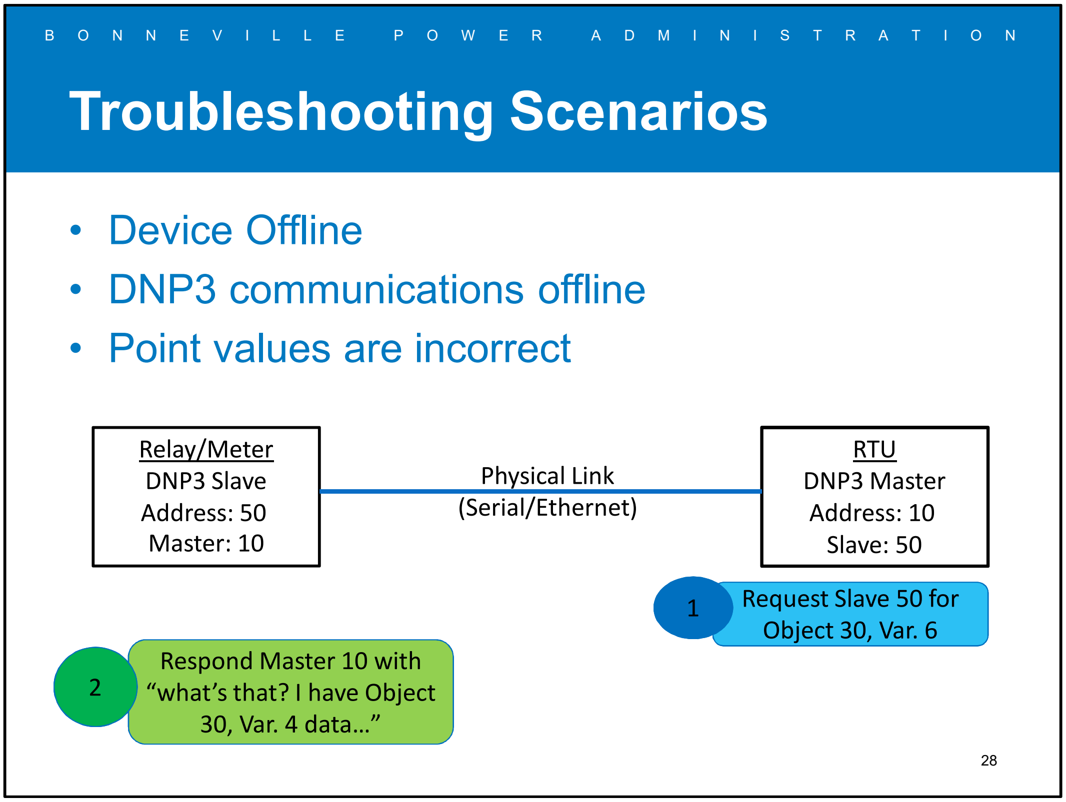
Table 3. Binary Input Points

Binary Input Points
Static Object Number: 1
Change Event Object Number: 2
Request Function Codes Supported: 1 (read)
Static Variation Reported When Variation 0 Requested: 1 (Binary Input Without Status)
Change Event Variation Reported When Variation 0 Requested: 2 (Binary Input Change With Time)

Point Index	Description	Change Event Assigned Class (1,2,3 or none)	Notes
0	50TA Phase Tripped	1	
1	50TB Phase Tripped	1	
2	50TC Phase Tripped	1	
3	150TA Phase Tripped	1	
4	150TB Phase Tripped	1	

27

The device manuals specify the objects and variations of the DNP3 data available in a device.

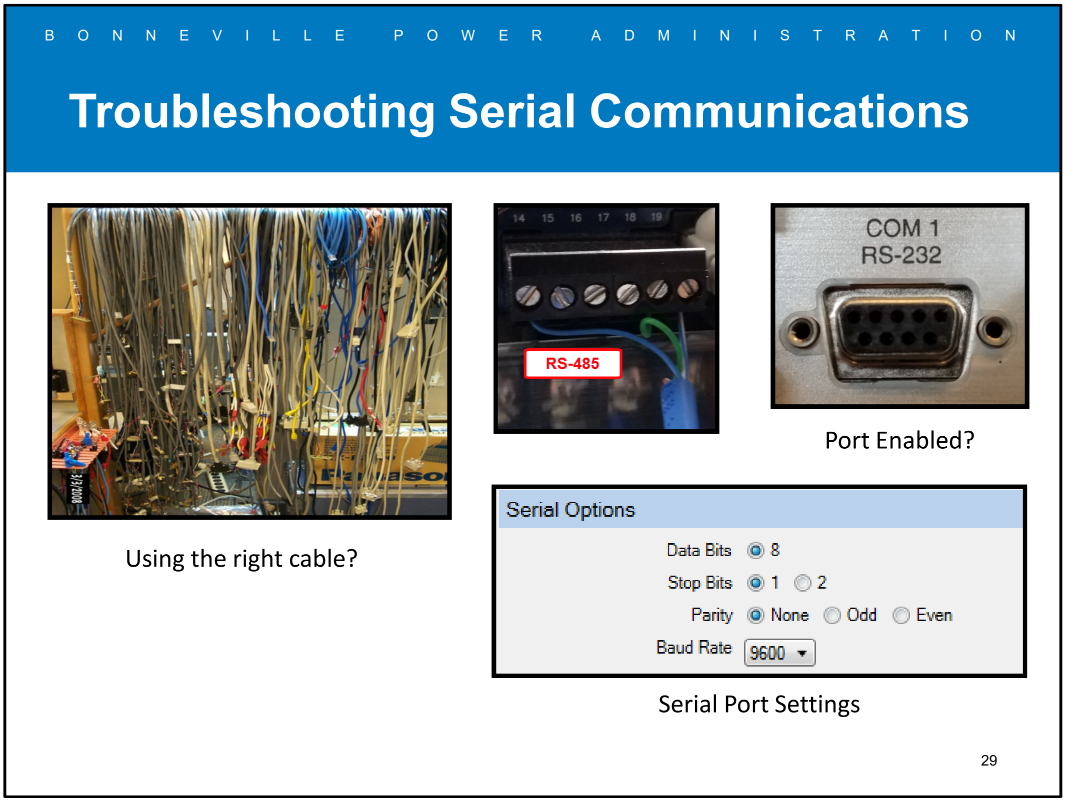


Common Errors:

- Physical Media
  - Device is not communicating at all.

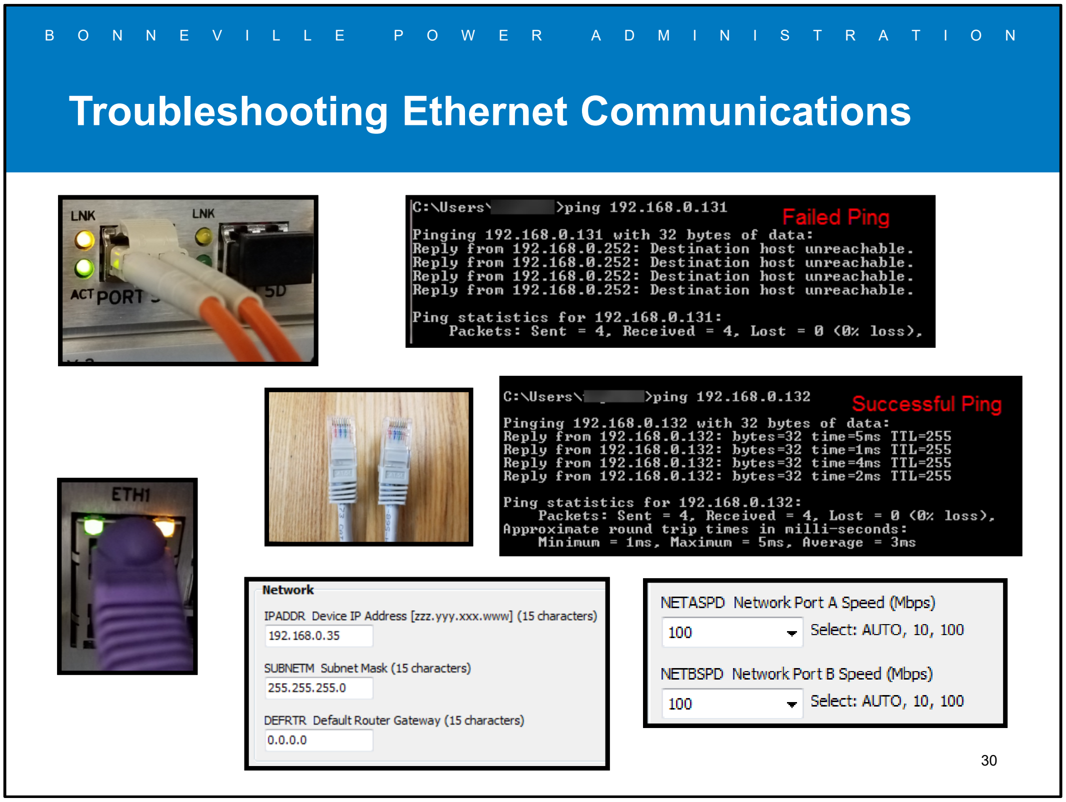
- Addressing
  - Device is not communicating but I see blinky lights on the port.

- Point Maps / Polling Strategy
  - Device is communicating, but the data looks incorrect or is offline.
  - The data exchange between the master and slave DNP3 devices only works if the slave understands the request and the master understands the response.
  - For example, if the master requests more points than what is available in the slave map, the slave won't know how to respond.
  - Data mismatch – data types that aren't supported. For example, if the master requests an object/variation that is not supported in the slave device.



For serial links, you may need to consider:

- Cabling (RS-232, RS-485, manufacturer specific cabling, etc.)
  - Null modem vs. straight through cable
- Enabling the correct serial port.
- Baud rate
- Data Bits – define how many bits are used in a single data transmission.
- Parity Bit – used for error detection
- Stop bit – used to indicate the end of a data transmission
- Handshaking / Flow Control – used to establish communication paths before data can be sent to avoid buffer overflow issues.



When troubleshooting Ethernet communications:

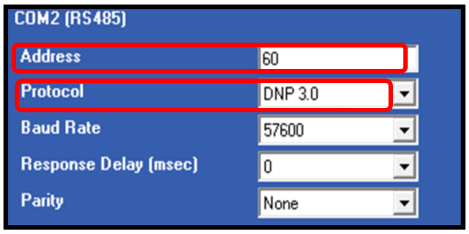
- Cabling (straight-through vs. crossover)
- For fiber, is TX or RX rolled?
- LEDs: Amber = Link Up, Green = Activity (TX/RX)
- Can you ping the device?
- Correct IP address, subnet mask, default gateway, speed/duplex (sometimes the speed and duplex is fixed).
- **Enabling the correct protocol**.

Example of speed/duplex mismatch:
Sometimes, managed Ethernet switches (i.e. configurable switches) are used where each switch port can be independently configured. You may have some baseline configuration where every port on the switch is configured with the same settings, including speed and duplex. Some devices that connect to the Ethernet switch may have a fixed speed and duplex for its Ethernet port that doesn’t match the switch port settings. It’s important to verify that the speed and duplex is consistent from sending device, through the LAN or WAN, to the receiving device and vice versa. This includes verifying any switches and/or routers.

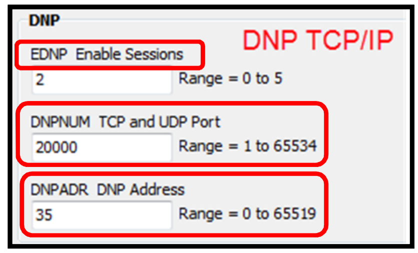
B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

DNP3 Link and Addressing

Serial DNP3 Slave



DNP3 TCP/IP Slave



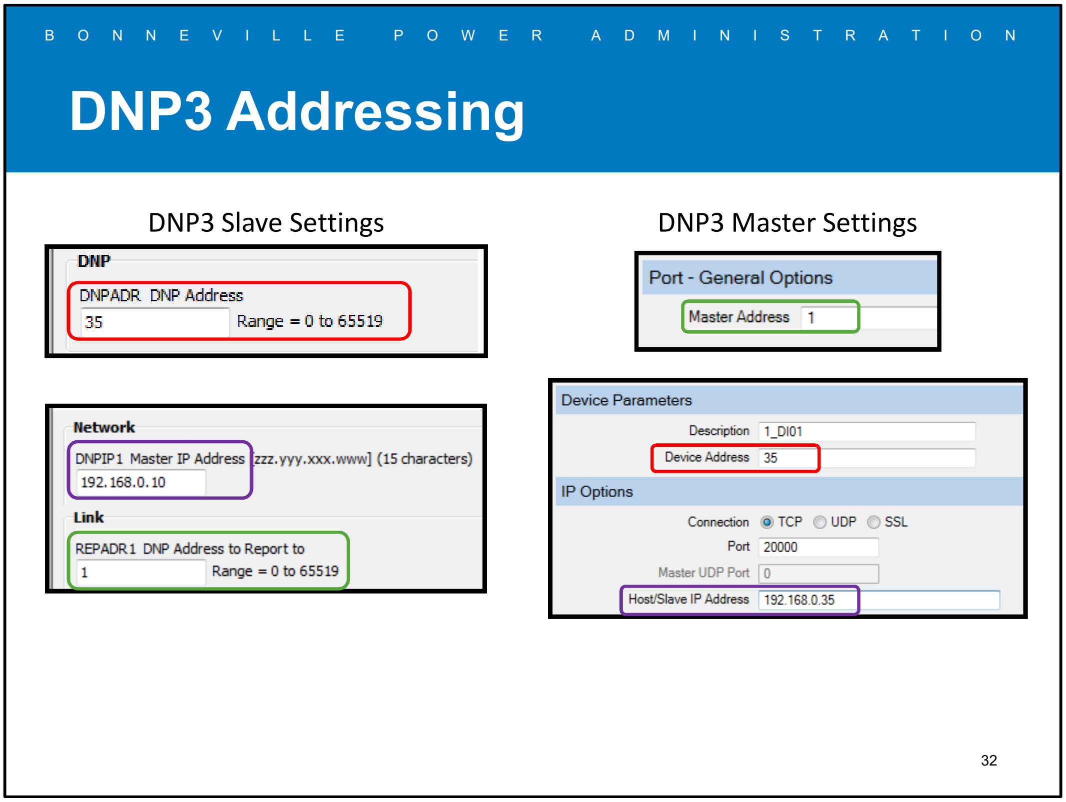
31

If all of the port settings look good and the device is still not communicating, move onto checking the DNP3 link settings.

Typically DNP3 needs to be enabled on the communications port that is being used.

For DNP3 TCP/IP, the slave device may be able to have multiple DNP3 sessions (i.e. report to multiple DNP3 masters), so you may need to specify how many sessions you want to enable. You also need to assign which TCP port to use (the TCP port assigned to DNP3 is 20000), which identifies which protocol is being used.

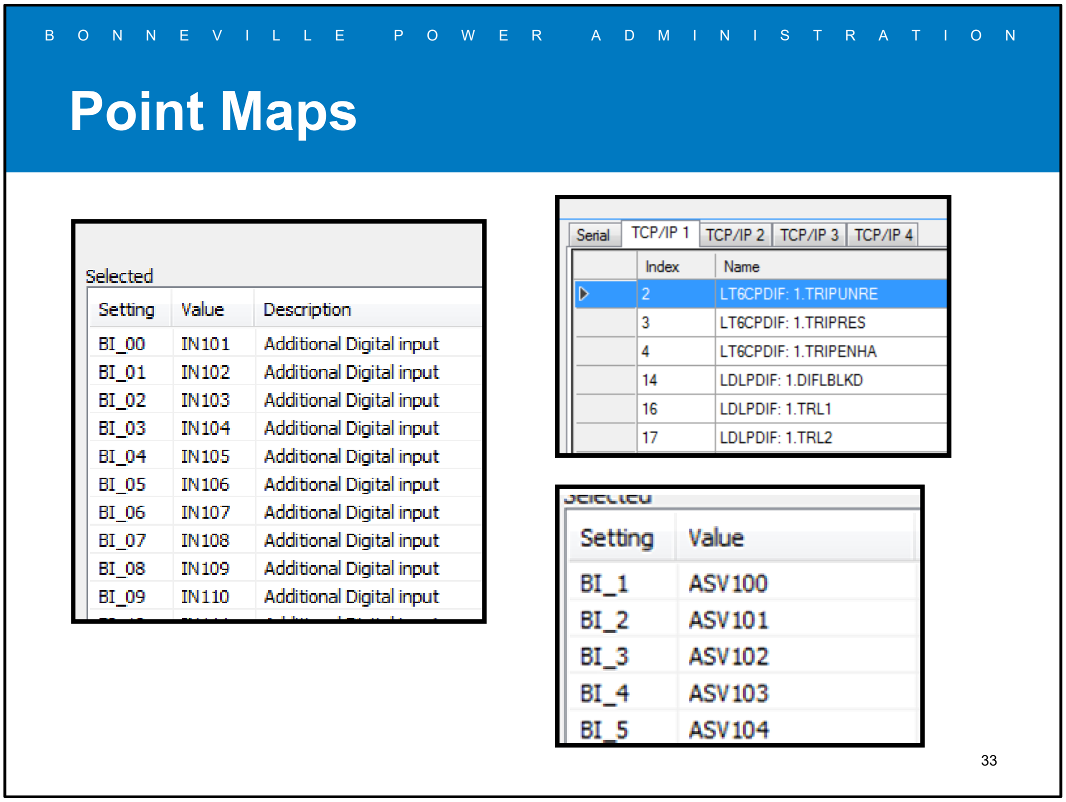
Each device needs a unique DNP3 address.



Slave devices need to know the master's DNP3 address.

Master devices need to know each slave's DNP3 address.

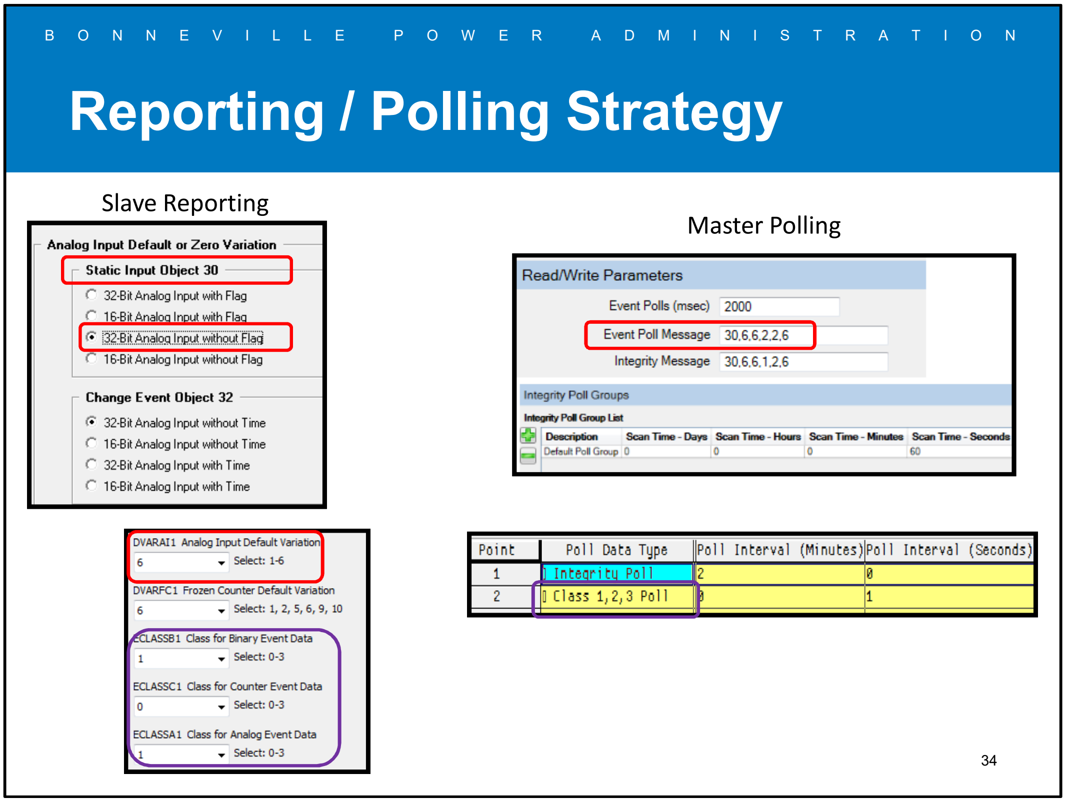
For DNP3 TCP/IP, each device needs to know each other's IP address as well.



If your data is offline or is not reporting properly, it is very important that point maps line up between the master and slave device so that when the master polls for a particular point index, it is receiving the correct value of that index. You can see how if the master polls for more points than what is configured, you will get errors because the slave device doesn't have enough points to respond with.

Here's a gotcha - some devices have DNP3 map indexes where the software shows the maps as 0 based, some are 1 based, some software doesn't even keep your map in sequential order. This is especially confusing when you are troubleshooting.

Another gotcha if you have rolled points in your point maps (similar to if you had rolled alarm wiring).



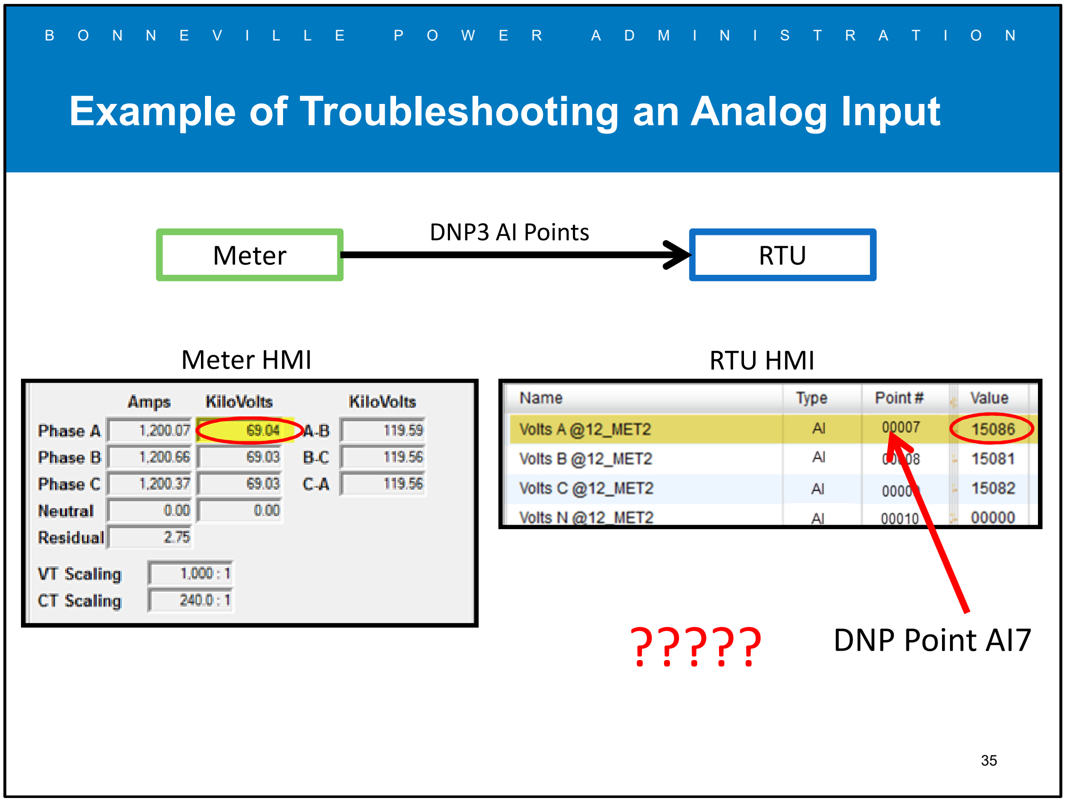
From relay to relay, it can vary what data is available in DNP3 and in what object/variation that data will be represented as.

What objects and variations does the master expect?

Does the IED support these objects and variations?

Is the proper event data assigned to the appropriate class?

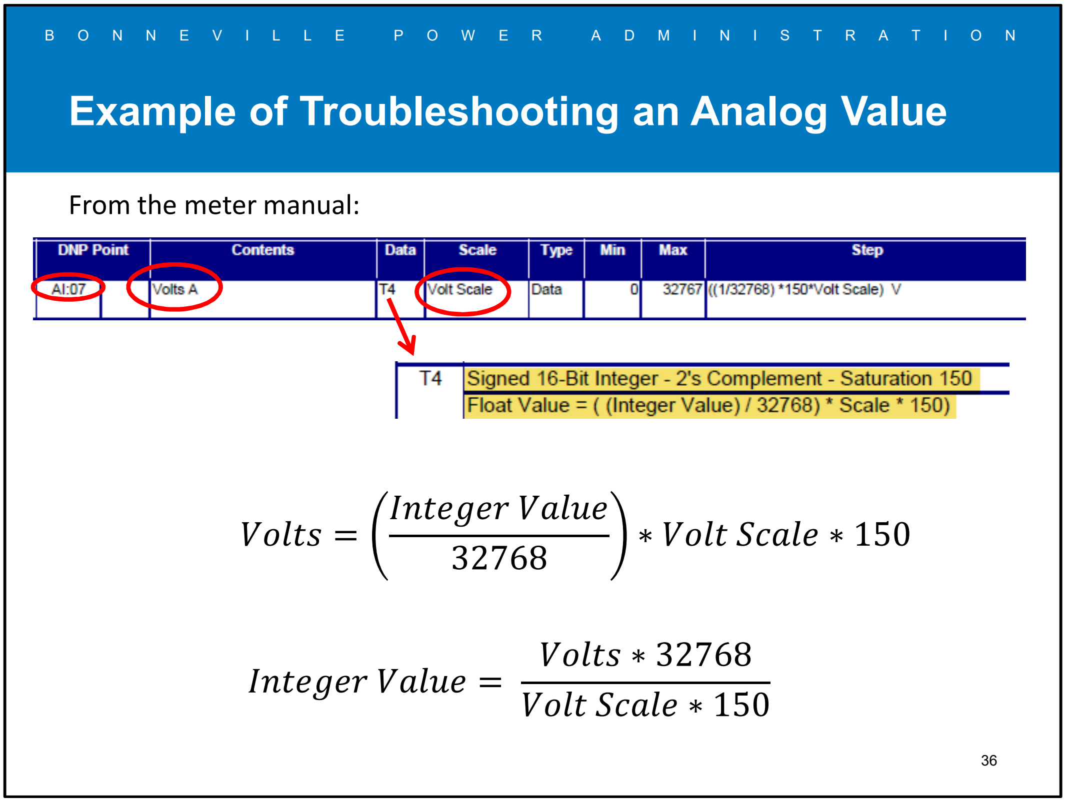
Typically the device manuals specify which DNP3 objects and variations are supported.



As an example, let's say you have a meter that is sending DNP3 analog inputs to your SCADA RTU.

Your meter is showing 69.04kV, but your SCADA RTU HMI is getting 15086. Why is this?

Let's take a look at what the manual says for DNP Analog Input point 7.



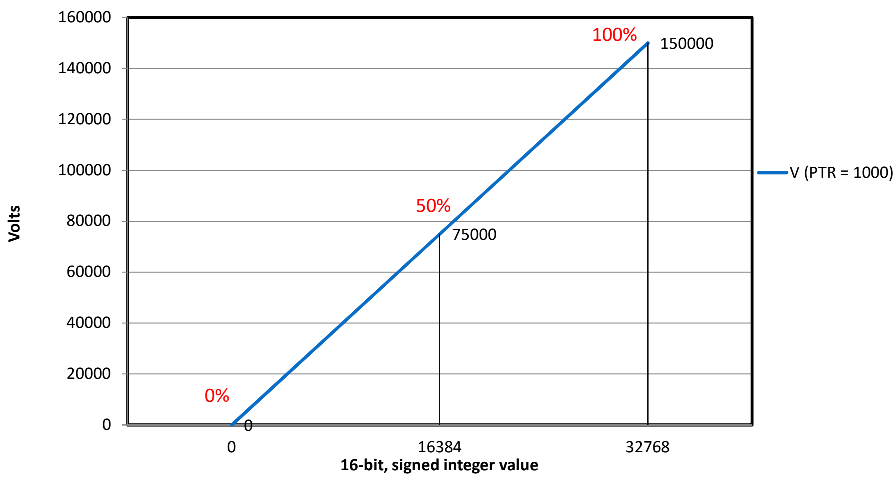
According to the manual's DNP map, AI:07 contains A phase voltage and is represented as a signed, 16-bit integer.

In order to calculate what the value is in units of volts or the float value, you have to apply the following equation to scale the value between a float value and an integer value. There is a saturation limit of 150 and the scale in this case is the volt scale.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

Example of Troubleshooting an Analog Value

$$Volts = \left(\frac{Integer\ Value}{32768}\right) * Volt\ Scale * 150$$



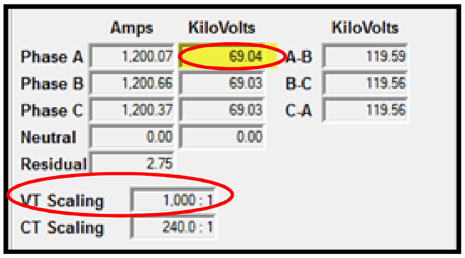
37

All this equation is doing is scaling the integer value into volts or vice versa.

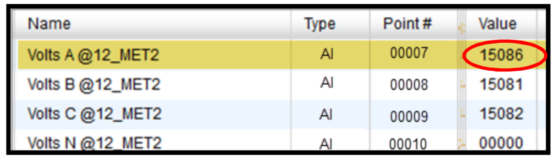
You can also think of it in terms of a percentage of a full scale where your scale of the 16-bit, signed integer representation goes from 0 to 32768 and you voltage scale goes from 0 to Volt Scale * 150 or 150,000 in this example.

Analog Value Example

Meter HMI

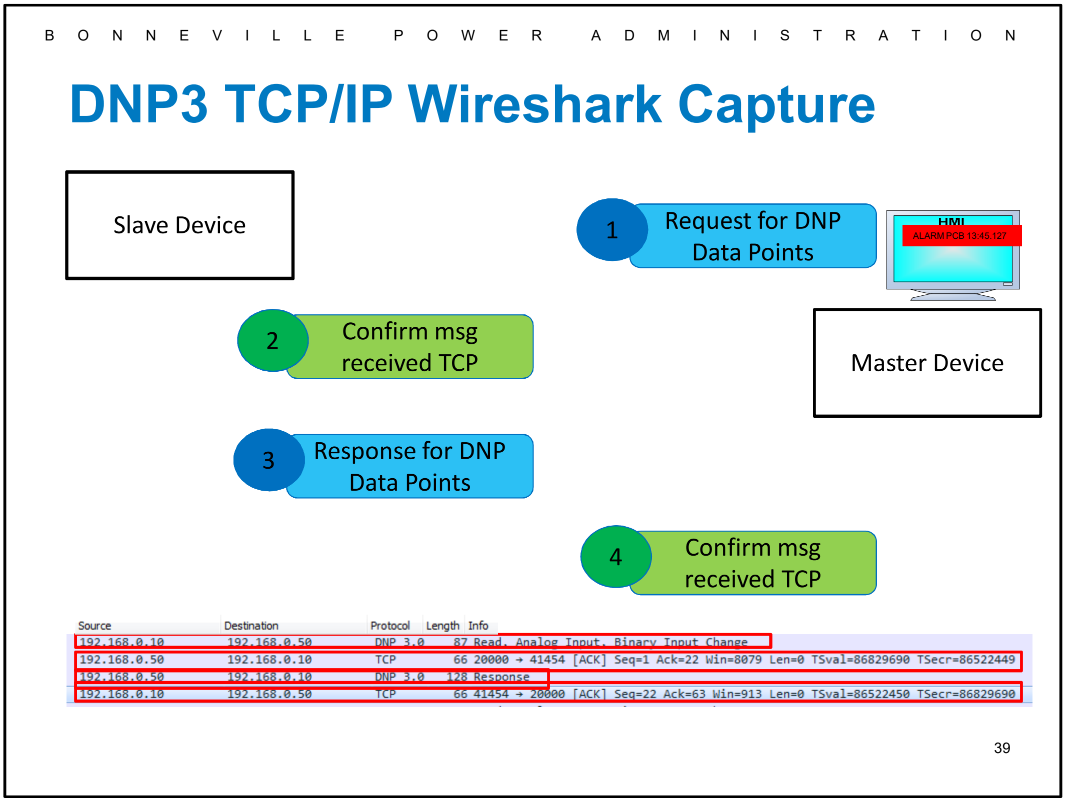


RTU HMI



$$Volts = \left(\frac{Integer\ Value}{32768}\right) * Volt\ Scale * 150$$

$$Volts = \left(\frac{15086}{32768}\right) * 1000 * 150 = 69{,}058.22V\ or\ \mathbf{69.06kV}$$



*Animated Slide

By monitoring DNP3 communications between two devices, it is easier to spot configuration errors and verify the devices are communicating appropriately.

NOTE: Same exchange happens for Modbus TCP/IP.

For every IED that reports to the Master, the following exchange happens:

1. Master sends a DNP3/Modbus request for data points following the standard polling strategy.
2. The IED confirms receipt of the TCP message.
3. The IED sends a DNP/Modbus response of data points.
4. The master confirms receipt of the TCP message.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

DNP3 TCP/IP Traffic Monitoring

```
▷ Frame 3603: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0
▷ Ethernet II, Src: Schweitz_0a:58:68 (00:30:a7:0a:58:68), Dst: Versalog_b0:50:c0 (00:04:bf:b0:50:c0)
▷ Internet Protocol Version 4, Src: 192.168.0.80, Dst: 192.168.0.10
▷ Transmission Control Protocol, Src Port: 20000, Dst Port: 53235, Seq: 1426, Ack: 323, Len: 146
◢ Distributed Network Protocol 3.0
  ▷ Data Link Layer, Len: 125, From: 80, To: 1, PRM, Unconfirmed User Data
  ▷ Transport Control: 0xcd, Final, First(FIR, FIN, Sequence 13)
  ▷ Data Chunks
  ▷ [1 DNP 3.0 AL Fragment (119 bytes): #3603(119)]
  ◢ Application Layer: (FIR, FIN, CON, Sequence 8, Response)
    ▷ Application Control: 0xe8, First, Final, Confirm(FIR, FIN, CON, Sequence 8)
      Function Code: Response (0x81)
    ▷ Internal Indications: 0x0000
    ◢ RESPONSE Data Objects
      ◢ Object(s): Binary Input Change With Time (Obj:02, Var:02) (0x0202), 2 points
        ▷ Qualifier Field, Prefix: 1-Octet Index Prefix, Range: 8-bit Single Field Quantity
        ▷ Number of Items: 2
        ◢ Point Number 0 (Quality: Online), Value: 1, Timestamp: Jun 19, 2018 18:56:20.551000000
            Index (8 bit): 0
          ◢ Quality: Online
              1... .... = Point Value: Set
              .0.. .... = Reserved: Not set
              ..0. .... = Chatter Filter: Not set
              ...0 .... = Local Force: Not set
              .... 0... = Remote Force: Not set
              .... .0.. = Comm Fail: Not set
              .... ..0. = Restart: Not set
              .... ...1 = Online: Set
            Timestamp: Jun 19, 2018 18:56:20.551000000 UTC
        ▷ Point Number 0 (Quality: Online), Value: 0, Timestamp: Jun 19, 2018 18:56:20.555000000
      ◢ Object(s): 64-Bit Floating Point Input (Obj:30, Var:06) (0x1e06), 10 points
        ◢ Qualifier Field, Prefix: None, Range: 8-bit Start and Stop Indices
            .000 .... = Prefix Code: None (0)
            .... 0000 = Range Code: 8-bit Start and Stop Indices (0)
        ▷ [Number of Items: 10]
        ▷ Point Number 0 (Quality: Online), Value: 0
        ▷ Point Number 1 (Quality: Online), Value: 0
        ▷ Point Number 2 (Quality: Online), Value: 0
```

40

Here is an example of a DNP3 TCP/IP response. You can see the response data objects are Object 02, Var. 02 and Object 30, Var. 06.

It can be used to verify that the master device has the proper request and the slave has the proper response.

Testing Inputs

IED Binary Input Map

Index	Value
0	52A
1	HWRALARM
…	…
N	

DNP3 Master Database

Index	Value
0	1
1	0
…	…
N	0

IED Analog Input Map

Index	Value
0	VA_MAG
…	…
N	

DNP3 Master Database

Index	Value
0	115.21
…	…
N	

Verify the value of each DNP3 input in the IED and compare it to the value received by the DNP3 master.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

Testing Digital Input

Relay Data

#	DATE	TIME	ELEMENT	STATE
20	06/06/2018	18:08:41.1401	RELAY ACCESS	ALARM
19	06/06/2018	18:08:41.1921	RELAY ACCESS	NORMAL
18	06/06/2018	18:08:48.3631	Relay	Disabled
17	06/06/2018	18:08:48.5256	Settings changed	Class A 2
16	06/06/2018	18:08:48.5256	Relay	Enabled
15	06/06/2018	18:08:48.5256	NOT LINK 5D	ALARM
14	06/06/2018	18:08:48.5256	CHANNEL 1 COM ALARM	NORMAL
13	06/06/2018	18:08:48.5256	CHANNEL 2 COM ALARM	NORMAL
12	06/06/2018	18:08:48.5796	CHANNEL 1 COM ALARM	ALARM
11	06/06/2018	18:08:48.5796	CHANNEL 2 COM ALARM	ALARM
10	06/14/2018	20:52:50.7841	RELAY ACCESS	ALARM
9	06/14/2018	20:52:50.8341	RELAY ACCESS	NORMAL
8	06/14/2018	21:03:26.5801	RELAY ACCESS	ALARM
7	06/14/2018	21:03:26.6321	RELAY ACCESS	NORMAL
6	06/14/2018	22:59:51.4611	PHASE_C Fault	Drop out
5	06/14/2018	22:59:51.4611	PHASE B Fault	Drop out
4	06/14/2018	22:59:51.4611	PHASE A Fault	Drop out
3	06/14/2018	23:00:25.6921	RELAY ACCESS	ALARM
2	06/14/2018	23:00:25.7421	RELAY ACCESS	NORMAL
1	06/14/2018	23:00:40.6566	Settings changed	Class P 5

=>

RTU Data

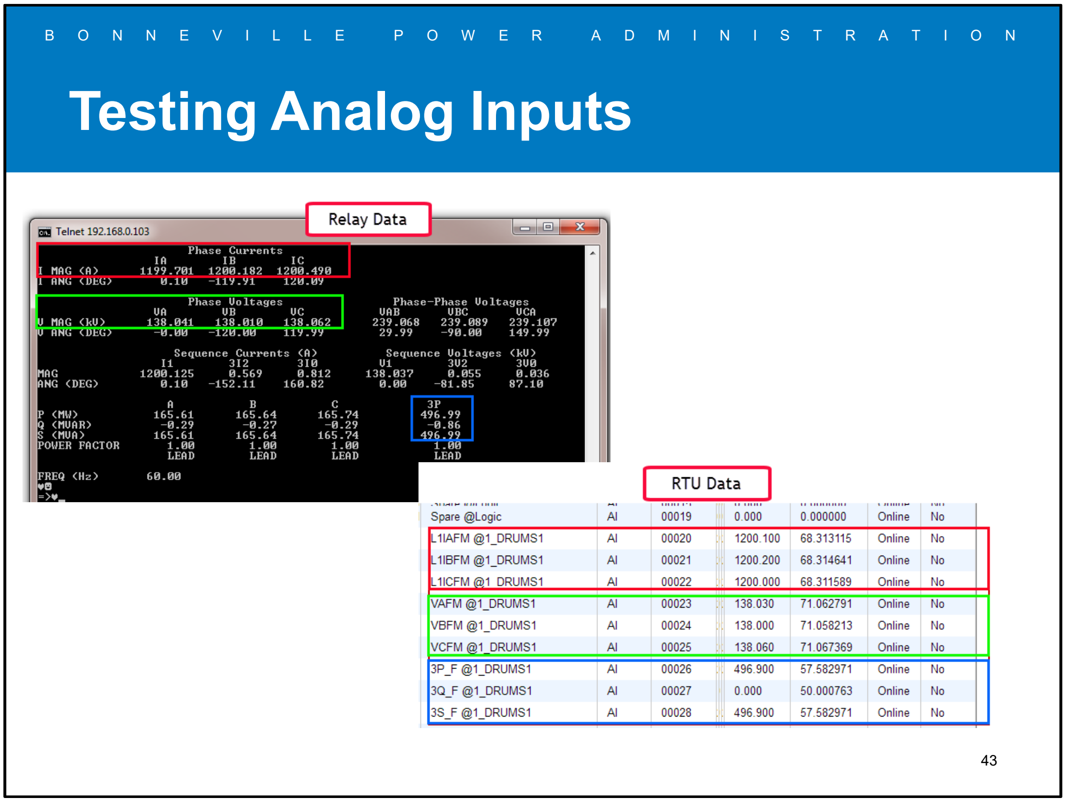
Device DateTime	Point Name	Alias	Value	Message
2018-06-14 23:00:25.742-07	ASV105 @SER__1_SEL411L	SER0082 [10H LINE] RELAY SET # RELAY ACCESSED	0	NORMAL
2018-06-14 23:00:25.692-07	ASV105 @SER__1_SEL411L	SER0082 [10H LINE] RELAY SET # RELAY ACCESSED	1	ALARM
2018-06-14 21:06:01.554-07	ASV100 @SER__1_SEL411L	SER0077 [10H LINE] RELAY SET # FAILURE NON-CRITICAL	0	NORMAL
2018-06-14 21:04:54.79-07	ASV100 @SER__1_SEL411L	SER0077 [10H LINE] RELAY SET # FAILURE NON-CRITICAL	1	ALARM
2018-06-14 21:03:26.632-07	ASV105 @SER__1_SEL411L	SER0082 [10H LINE] RELAY SET # RELAY ACCESSED	0	NORMAL
2018-06-14 21:03:26.58-07	ASV105 @SER__1_SEL411L	SER0082 [10H LINE] RELAY SET # RELAY ACCESSED	1	ALARM
2018-06-14 20:52:50.834-07	ASV105 @SER__1_SEL411L	SER0082 [10H LINE] RELAY SET # RELAY ACCESSED	0	NORMAL
2018-06-14 20:52:50.784-07	ASV105 @SER__1_SEL411L	SER0082 [10H LINE] RELAY SET # RELAY ACCESSED	1	ALARM
2018-06-07 19:42:56.321-07	Comm Fail @SER__1_REAC1S1	SER0152 COMM FAIL	1	ALARM
2018-06-07 19:40:25.168-07	SV11 @SER__1_REAC1S1	SER0162 REACTOR TRIP	0	NORMAL
2018-06-07 19:39:33.26-07	SV11 @SER__1_REAC1S1	SER0162 REACTOR TRIP	1	ALARM

42

Microprocessor based devices typically have interfaces or terminal commands that can be used to read data values.

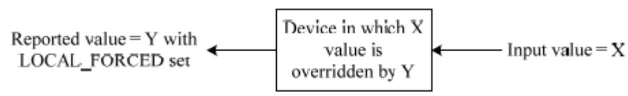
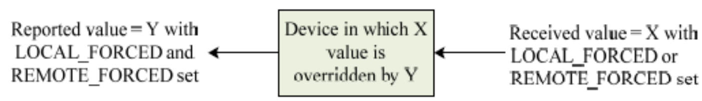
This can be used to confirm that a value transmitted by the IED gets received and decoded appropriately in the master or vice versa.

This is an example of verifying digital input events.



Example of verifying analog inputs.

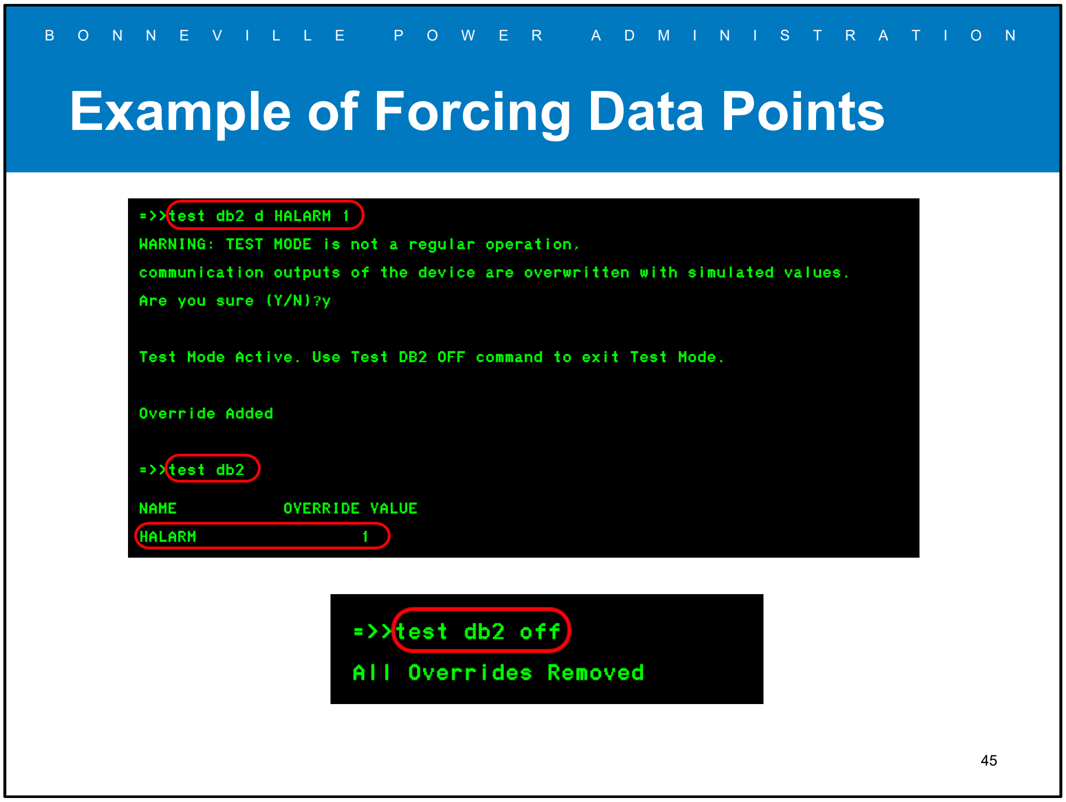
Forcing Data Points

Name	Functional description
LOCAL_FORCED	If set, the data value is overridden by the device that reports this flag as set. This may be due to the device operating in a diagnostic or temporary mode or due to human intervention.  If the value is forced in a non-originating device and overridden in a downstream device, then the non-originating device shall set both REMOTE_FORCED and LOCAL_FORCED flags.  See flag description 11.6.1.1, NOTE 3.

44

Some DNP3 slave devices allow the “LOCAL_FORCED” DNP3 flag to be set. Flags are only available in certain DNP3 object variations.

Using the LOCAL_FORCED flag, you can overwrite values so that the device will report the forced value. This is especially useful for doing a quick check that the DNP3 maps between the IED and the RTU are configured properly.



An example is the SEL 400 series relays use the “test db2” command to set the LOCAL_FORCED flag in DNP3 in order to overwrite values in their “db2” database.

B O N N E V I L L E   P O W E R   A D M I N I S T R A T I O N

Testing Outputs

IED Binary Output Map

Index	Value
0	OUT101:OUT102
1	OUT103:OUT104
…	…
N	OUTxxx:OUTyyy

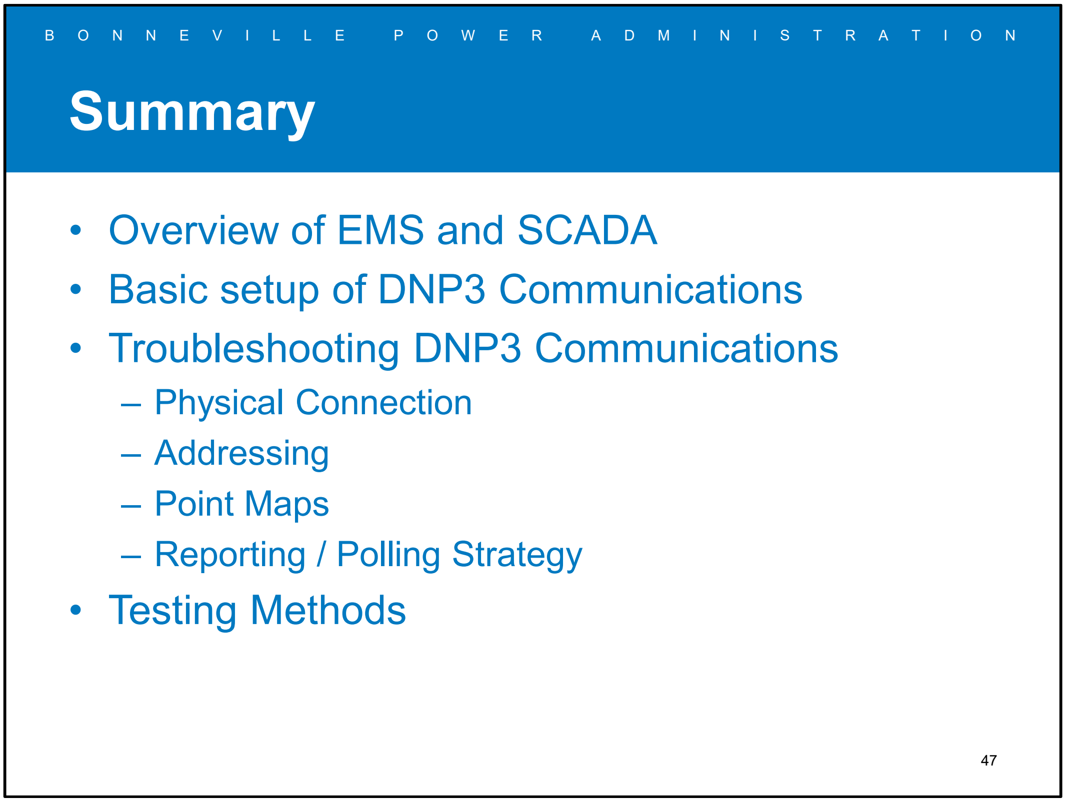
DNP3 Master Command

Index	Value	On Time
0	TRIP	2 seconds

Verify OUT101 pulses for 2 seconds.

An example of testing the binary output map may be to have the DNP3 master send a binary output command to the slave device and verify the slave device receives the command and takes action appropriately.

In this example, there are two outputs per binary output point – one that will operate for a trip command and another that will operate for a close command. If the DNP3 master sends a trip command with an on-time of 2 seconds, you should expect the appropriate output to pulse for the on-time duration.



As a summary, I gave an overview of what is EMS and what is SCADA.

Then we covered a basic setup of DNP3 communications – specifically using the example of a relay being a DNP3 slave and an SCADA RTU being a DNP3 master.

Finally, we went over how to troubleshoot DNP3 communications. Depending on the issues you encounter:

- Physical Connection – make sure your serial or Ethernet settings and physical connections are correct.
- Addressing – make sure your DNP3 addressing and/or IP addressing is correct.
- Point Maps – when testing the point maps, it's important that the master know the point map of the slave device exactly.
- Reporting / Polling Strategy – knowing how the master should poll the slave and how the slave should respond will aid in troubleshooting.

Questions?