



Handling Modbus in an AB World

Lynn A Linse
Principal Engineer



Linking Modbus and AB

- Multiple ways to do on the market
 - AB PLC with Modbus driver
 - Add PLC rack card with Modbus
 - Use “virtual PLC” supporting 2 protocols
 - Digi One IAP’s “Protocol Irrelevance”
- Cover the Pros and Cons of each
- Demonstrate Digi One IAP’s approach



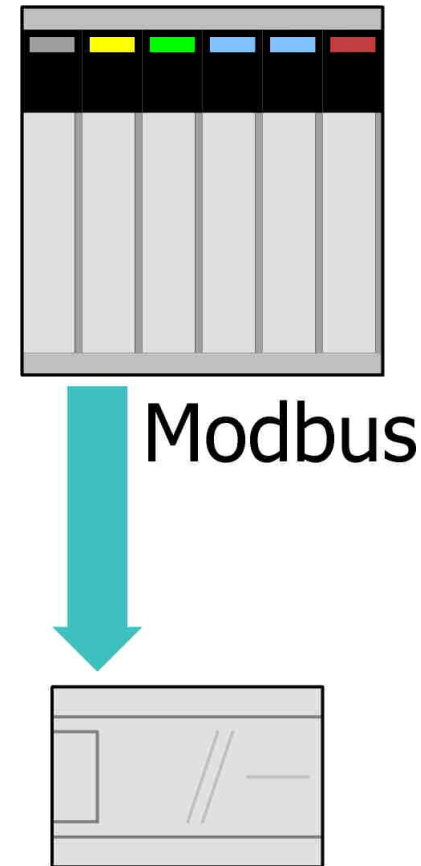
Modbus “Staying Power”

- Modbus has been around for over 25 years; is older than other common protocols
- Modbus prospers because:
 - Drivers can be written literally in a few hours; complex “modern” protocols like Ethernet/IP, ProfiNET, or IDA require months to write
 - Modbus forces no specific design on data
 - Massive list of interoperable products
 - Easy to emulate in BASIC and hard-code into PLC “downloadable code” modules



AB PLC with Serial Modbus

- MicroLogix 1100, 1200 & 1500 support Modbus driver directly
- DF1 is disabled





AB PLC with Serial Modbus

Pro

- Is “Free”; nothing extra to buy ... maybe

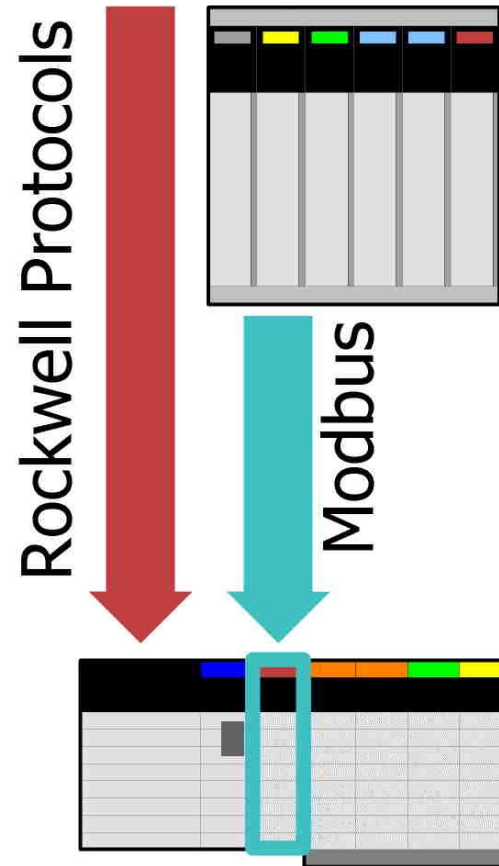
• Con

- Locks out RSLinx & RSLogix access
- Must reboot PLC to re-enable Rockwell protocol access temporarily
- May cause you to “buy-up” to a 2-port model to avoid loss of RSLinx support



AB PLC with Rack Card

- Add Modbus serial or Ethernet port to AB PLC Rack
- Allows concurrent Modbus and Rockwell access





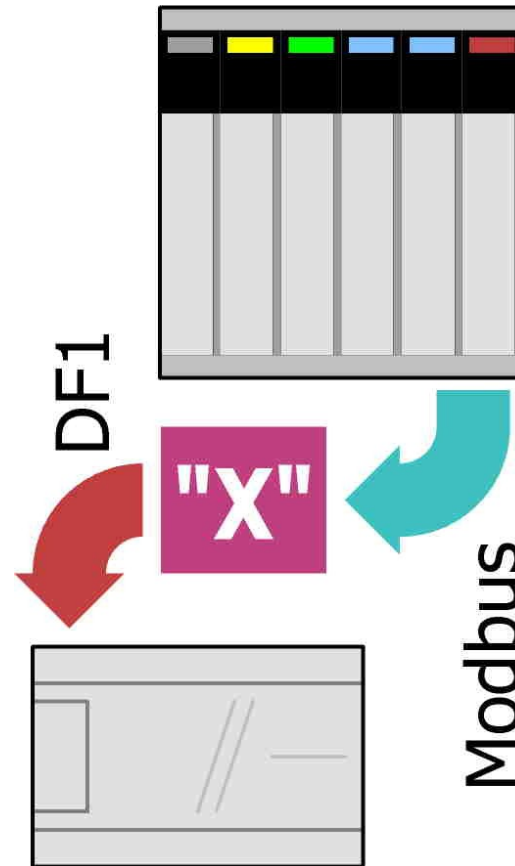
AB PLC with Rack Card

- Pros
 - Well established paradigm; it works
- Cons
 - High cost due to custom hardware, custom engineering and limited volumes
 - PLC program must change
 - Consumes rack slot and backplane power
 - Requires 2nd Ethernet cable and resource
 - Result is not transparent; Polling a Modbus register tells nothing about AB data it represents



Virtual PLC Solution

- “Magic Box”
 - Modbus on one side
 - AB on the other
- Limited or no direct Rockwell protocol access allowed





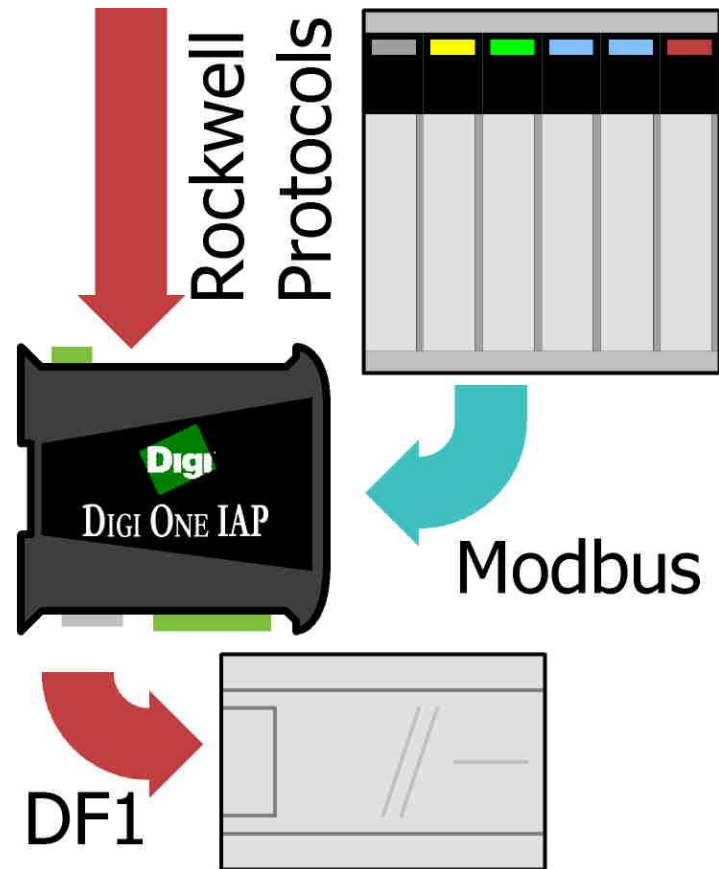
Virtual PLC Solution

- Pros
 - Cost likely lower than rack card
 - Well established paradigm
 - “Cache” behavior for multiple masters
 - Unlimited ability to massage & rearrange data
- Cons
 - Requires explicit configuration to define mapping
 - Future configuration changes hard to detect by old masters
 - Detecting slave failure requires reading special registers
 - Blocks RSLinx/RSLogix access to AB PLC serial port



Digi “Protocol Irrelevance”

- Modbus requests are handled using appropriate Modbus or AB slave protocols
- AB requests are handled using appropriate Modbus or AB slave protocols





Digi “Protocol Irrelevance”

- Radical new approach.
- Is transaction oriented.
 - Digi One IAP has no concept of data and no retained data memory.
 - Each request received creates one transaction to the slave with the appropriate protocol and values.
- Network or serial masters can query any mix of network or serial slaves.



Traditional AB Bridging

- Ethernet/IP querying AB/DF1 with DOIAP
 - Ethernet/IP request is received.
 - Bridge examines request details (connection path) to learn which slave this request targets.
 - Bridge extracts the core PCCC request – this is the SLC5 or PLC5 style read/write command.
 - Bridge rewraps PCCC as CSPv4 or DF1 request.
 - When response is received, Bridge reverses the process to reconstruct Ethernet/IP response.
- Bridge works because both Master & Slave understand the PCCC message structure.



Traditional Modbus Bridging

- Modbus/TCP querying Modbus/RTU with DOIAP
 - Modbus/TCP request is received.
 - Bridge examines request details (Unit Id) to learn which slave this request targets.
 - Bridge extracts the core Modbus ADU – this is the read/write holding registers and other commands.
 - Bridge rewraps Modbus ADU as Modbus/RTU.
 - When response is received, Bridge reverses the process to reconstruct Modbus/TCP response.
- Bridge works because both Master & Slave understand the Modbus ADU structure.



AB reading Modbus

- Ethernet/IP querying Modbus/RTU
 - Ethernet/IP request is received.
 - DOIAP examines request details (connection path) to learn which slave this request targets.
 - DOIAP extracts the core PCCC request – this is the SLC5 or PLC5 style read/write command.
 - DOIAP detects slave does NOT understand PCCC.
 - DOIAP examines PCCC; breaks down request
 - Commands reduced to (example) read twenty words
 - Addresses such as N7:20 or N50:0 are converted by formula to a virtual offset – 20 and 4000 respectively



AB reading Modbus (cont)

- Ethernet/IP querying Modbus/RTU (cont)
 - DOIAP queues generic request for Modbus slave driver.
 - Example: read twenty 16-bit words starting at offset 4000
 - Modbus driver opens request
 - Detects that it is NOT Modbus ADU
 - Generates Modbus ADU with same meaning
 - Example: becomes function 3 read starting at 4x04001
 - Modbus driver sends Modbus request to Slave
 - When response is received, DOIAP reverses the process:
 - Convert Modbus response into generic 20 word data response
 - Convert generic 20 word data response to PCCC
 - Returns PCCC to Ethernet/IP master



AB reading Modbus

- What did we accomplish?
 - ControlLogix thinks it is reading 20 elements starting at N50:0 from a SLC5 or MicroLogix
 - Modbus slave thinks it is being polled for 20 registers with Function 3 starting at 4x04001
- DOIAP does this conversion mechanically without explicit configuration
- Any Modbus data can be accessed given the appropriate SLC5/MicroLogix notation
 - N11:74 accesses register 4x00175
 - B10:14/6 accesses coil 0x00231
 - Excel spreadsheet shows correct AB notation, given Modbus



Digi Doc: 90000652.xls

Given Modbus 4x register notation,
what is the AB SLC500 file notation?

100

Enter Modbus here: 4x << Enter Slave Value Here

Absolute Word	SLC500 File Notation	Modbus Notation	
128	N11: 28	4x 00129	<< Use AB Notation
129	N11: 29	4x 00130	
130	N11: 30	4x 00131	F8 or F10 accesses 4x000001
131	N11: 31	4x 00132	Assumes Modicon format, so
132	N11: 32	4x 00133	4x000001 is LOW word of IEEE float
133	N11: 33	4x 00134	4x000002 is HIGH word of IEEE float
134	N11: 34	4x 00135	
135	N11: 35	4x 00136	F8:1 is 4x000003 + 4x000004
136	N11: 36	4x 00137	F8:2 is 4x000005 + 4x000006
137	N11: 37	4x 00138	and so on



User Documentation

AB	Modbus	Mode	Desc	Values
N10:0	4x0001	Rd	Drive Horsepower	In HP (ie: 15 = 15hp)
N10:1	4x0002	Rd	Software version	BCD like 0101
N10:2	4x0003	Rd/Wr	Parameter Lock	0/disable, 1/enable
N10:3 N10:3/0 N10:3/1 N10:3/2 N10:3/3 N10:3/4	4x0004	Rd/Wr	Run Commands bit 1: Stop / Brake bit 2: Forward Run bit 3: Reverse Run bit 4: Fault Reset bit 5: Coast to Stop	Only 1 bit may be set per write, or the write is ignored. Read back shows current status.
N10:4	4x0005	Rd	Current	By 0.1A
N10:5	4x0006	Rd	Frequency	By 0.01 Hz
N10:6	4x0007	Rd	Voltage	volts
N10:7	4x0008	Rd	Output Power	Watts
N10:8	4x0009	Rd	Speed	RPM



Modbus reading AB

- Modbus/TCP querying DF1 Full-Duplex
 - Modbus/TCP request is received.
 - DOIAP examines request details (Unit Id) to learn which slave this request targets.
 - DOIAP extracts the Modbus ADU.
 - DOIAP detects slave does NOT understand Modbus ADU.
 - DOIAP examines Modbus; breaks down request
 - For example: read twenty 16-bit words
 - Addresses such as 4x00175 or 4x04001 are converted by formula to a virtual offset – 174 and 4000 respectively



Modbus reading AB (cont)

- Modbus/TCP querying DF1 Full-Duplex (cont)
 - DOIAP queues generic request for DF1 Full-Duplex driver.
 - Example: read twenty 16-bit words starting at offset 12500
 - DF1 Full-Duplex driver opens request
 - Detects that it is NOT PCCC
 - Attempts to generate PCCC with same meaning
 - Example: becomes SLC5 read starting at N50:0
 - DF1 driver sends DF1 request to Slave
 - When response is received, DOIAP reverses the process:
 - Convert PCCC response into generic 20 word data response
 - Convert generic 20 word data response to Modbus
 - DOIAP returns Modbus to Modbus/TCP master



Modbus reading AB

- What did we accomplish?
 - Modbus/TCP master thinks it is polling 20 registers with Function 3 starting at 4x12501
 - Micrologix thinks it is being asked for 20 elements starting at N50:0 from another Rockwell PLC or application
- DOIAP does this conversion mechanically without explicit configuration
- Any PCCC data can be accessed given the appropriate Modbus notation
 - 4x02825 accesses integer N11:74
 - 0x08231 accesses bit B10:14/6
 - Excel spreadsheet shows correct Modbus notation, given AB



Digi Doc: 90000653.xls

9	You have Rockwell N-File notation,			
10	what is the Modbus Holding register notation?			
11			20	
12	Enter N-File: N	<input type="text" value="20"/>	<< Enter Slave Value Here	
13	Starting Element:	<input type="text" value="17"/>	<< Enter Slave Value Here	
14				
15	Absolute Word	SLC500 File Notation	Modbus Notation	
16	5017	N20: 17	4x 05018	4x00001/N0 treated as N7
17	5018	N20: 18	4x 05019	4x00251/N1 accesses I1 ReadOnly
18	5019	N20: 19	4x 05020	4x00501/N2 accesses S2 ReadOnly
19	5020	N20: 20	4x 05021	4x00751/N3 accesses B3 Read/Write
20	5021	N20: 21	4x 05022	4x01001/N4 accesses T4 ReadOnly
21	5022	N20: 22	4x 05023	4x01251/N5 accesses C5 ReadOnly
22	5023	N20: 23	4x 05024	4x01501/N6 accesses R6 ReadOnly
23	5024	N20: 24	4x 05025	4x01751/N7 accesses N7 Read/Write
24	5025	N20: 25	4x 05026	4x02001/N8 accesses F8 Read/Write
25	5026	N20: 26	4x 05027	4x63251/N253 accesses O0 ReadOnly
26	5027	N20: 27	4x 05028	All others must exist as N-file or
27	5028	N20: 28	4x 05029	Modbus Exception returned



AB/Modbus Data Details

- The “Rockwell polling Modbus” formulas are different than “Modbus polling Rockwell”
 - Due to limitations in first 9 files: for example N2 & B6 are not legal AB notation
- 16-bit words - AB to/from Modicon PLC
 - DOIAP swaps bytes
- 1-bit packed as words - AB to/from Modicon PLC
 - DOIAP must not swap bytes
- 32-bit long/float – AB to/from Modicon PLC
 - DOIAP swaps bytes, but not the words
- Even though Modbus is “Big-Endian”, Modicon PLC use x86 chips and are native “Little-Endian”
- All AB special files (status, counters, etc) are converted to Modbus words as-is – users must read AB docs to understand



Digi One IAP solution

- Pros
 - No data configuration required or allowed
 - What works today; works next year
 - Slave failure detected by failure to respond
 - 100% transparent pass-thru of native protocols
 - Concurrent access by Rockwell and Modbus
 - One firmware does all conversions
- Cons
 - No ability to rearrange or tweak data
 - No “cache” behavior – every request creates a new transaction



One Firmware does All

- Example Applications:
 - AB PLC polling Modbus slaves
 - Old PLC5E using Ethernet to poll ControlLogix
 - ControlLogix and Modbus/TCP acting as peers
 - Old “AB/Ethernet” (CSPv4) software polling ControlLogix
 - Polling single-port Micrologix as both Modbus/TCP and DF1 slave concurrently
 - Ethernet-enabling SLC5/03 or 04



Support Documentation

http://www.digi.com/support/supporttype.jsp?tp=3

Contact us | Part

Products Applications **Support** News About us Where to buy

Drivers
Firmware upgrades
Documentation
Diagnostics, Utilities and MIBs
Cabling
Online Support Request
Knowledge Base
Warranty Registration
Support Forums

Home > Support > Documentation

Documentation

To find the correct document for your product below.

Select Your Product for Search

Digi One IAP (Formerly IA RealPort)

Submit

- OR -

Identify Product by Entering...

- Part Number
- Product Keyword(s)
- ...



Support Documentation

Relevant Documentation at www.digi.com

- 90000643_a.pdf (Acrobat)
 - App Note; Rockwell polling Modbus
- 90000652_A.xls (Excel)
 - Calculator; Rockwell Master
- 90000647_B.pdf (Acrobat)
 - App Note; Modbus polling Rockwell
- 90000653_B.xls (Excel)
 - Calculator; Modbus Master



Digi One IAP

- **Full Feature Model**
- 10/100 HD/FD Ethernet
- 1 or 2 serial ports
- RS-232/422/485
- 10-30vdc Power Supply
- 2kv isolation between power and serial
- Div 1 Class II
- Protocols: Modbus, Rockwell, Omron, ASCII, user-defined
- Fall-2006:
 - 400-watt TVS diodes
 - Optional 802.3af Power-Over-Ethernet





Digi One IA

- ***Industrial single-port model***
- 10/100 HD/FD Ethernet
- 1 serial port
- RS-232/422/485
- 10-30vdc Power Supply
- 2kv isolation between power and serial
- Div 1 Class II
- Protocols: Raw or Limited Modbus Bridge (matches Modicon CEV specs)
- 400-watt TVS diodes (fall-2006)





Digi One SP-IA

- ***Lowest Cost DIN-rail***
- 10/100 HD/FD Ethernet
- 1 serial port
- RS-232/422/485
- 10-30vdc Power Supply
- No DC supply isolation
- Raw access only; no special protocols
- Office-grade surge protection





Digi Connect WiSP

- 802.11b WiFi
- 1 serial port
- RS-232/422/485
- 10-30vdc Power Supply
- No DC supply isolation
- Raw access only;
no special protocols
- Office-grade surge
protection
- Optional DIN-rail
bracket

