Overview:

The TAGWELL TLM is a free firmware add-on option for the SoftPLC runtime control software versions 4.6 and later that adds the capability of receiving and sending data messages to the TagWell Web Portal. This capability is implemented as a TOPDOC Loadable Module (TLM), written in C++ and implements a driver that receives data messages from TagWell and a ladder instruction (TLI) which queues data messages to be sent. This document describes the installation, usage, and functionality of the TAGWELL TLM.

Concepts:

The **SoftPLC runtime engine** software supports TLM's, which are shared library extensions to SoftPLC. A TLM may be loaded either as a **DRIVER** or as a **MODULE**. The difference between a DRIVER and a MODULE is that a DRIVER is called once per SoftPLC scan, and optionally an additional number of times per scan. A MODULE is only called when the control program decides to call it and not as an inherent part of the scan. TLM's are made known to SoftPLC in the MODULES.LST file which may be edited by **TOPDOC NexGen** by traversing to: PLC | Modules.

This TAGWELL TLM is to be used as a DRIVER, not as a MODULE. It looks for incoming messages every scan and it provides one (1) TOPDOC Loadable Instruction (TLI) to send a data message.

Features:

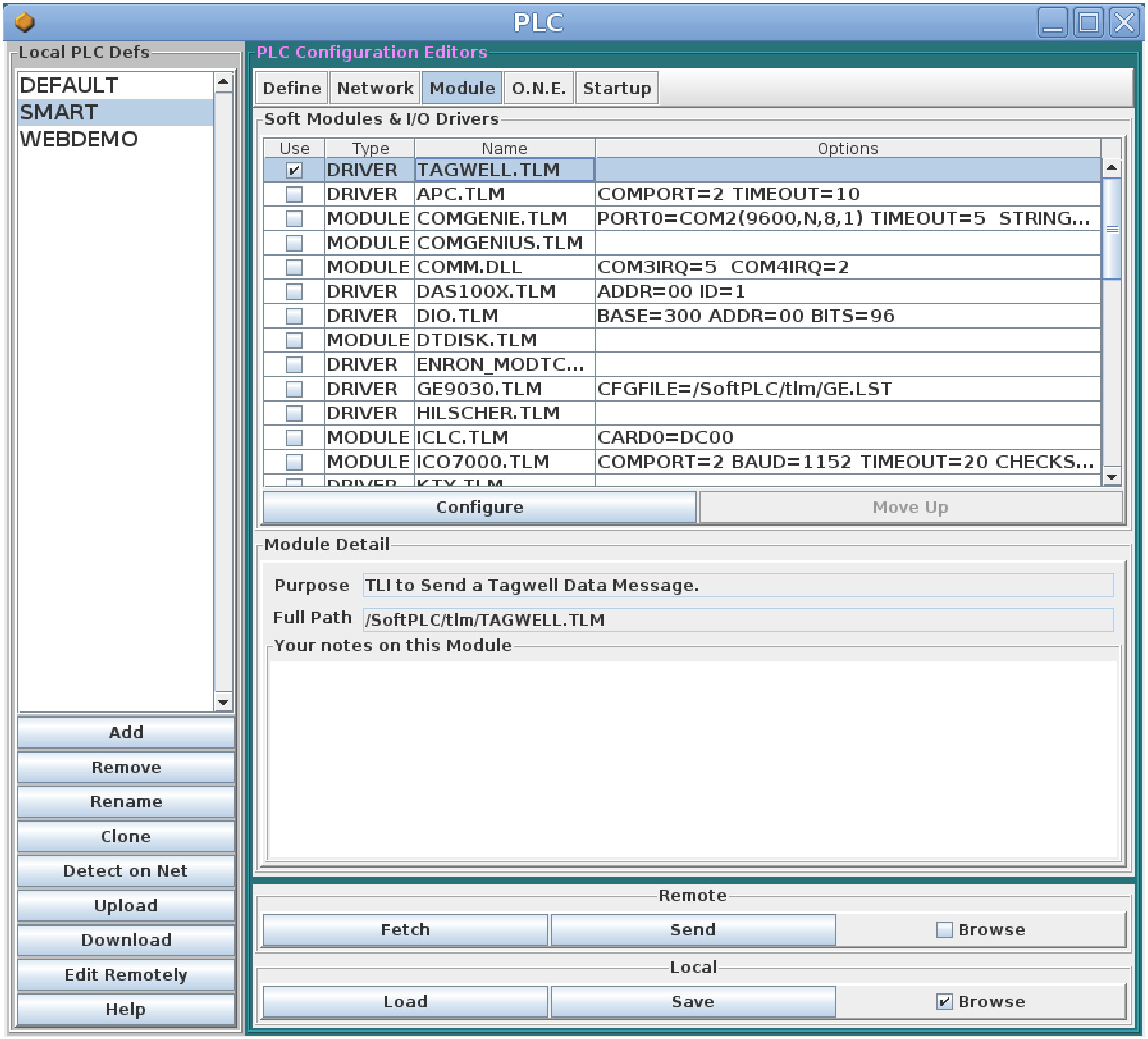
- Communicates with the TagWell Web Portal via cell modem on VPN or over Internet.
- Receives unsolicited messages from the TagWell Web Portal.
- TagWell shared data lists are created using Tags rather than data table addresses.
- Tag lists may be created using regular expressions. *(See* http://regexlib.com *for information on regular expressions.)*

Requirements:

- Gatecraft Linux version 3.6 or greater
- SoftPLC version 4.6 or greater
- TagWell Web Portal Account

Installation:

The TLM is named **tagwell.tlm.so**, and the configuration file is **TAGWELL.LST**. Both of these files will be pre-installed on your SoftPLC for you. To use it you simply have to enable and configure it using **TOPDOC NexGen's** PLC | MODULES editor. Select **Use** for TAGWELL.TLM, then click on the Configure button to edit the TAGWELL.LST file.



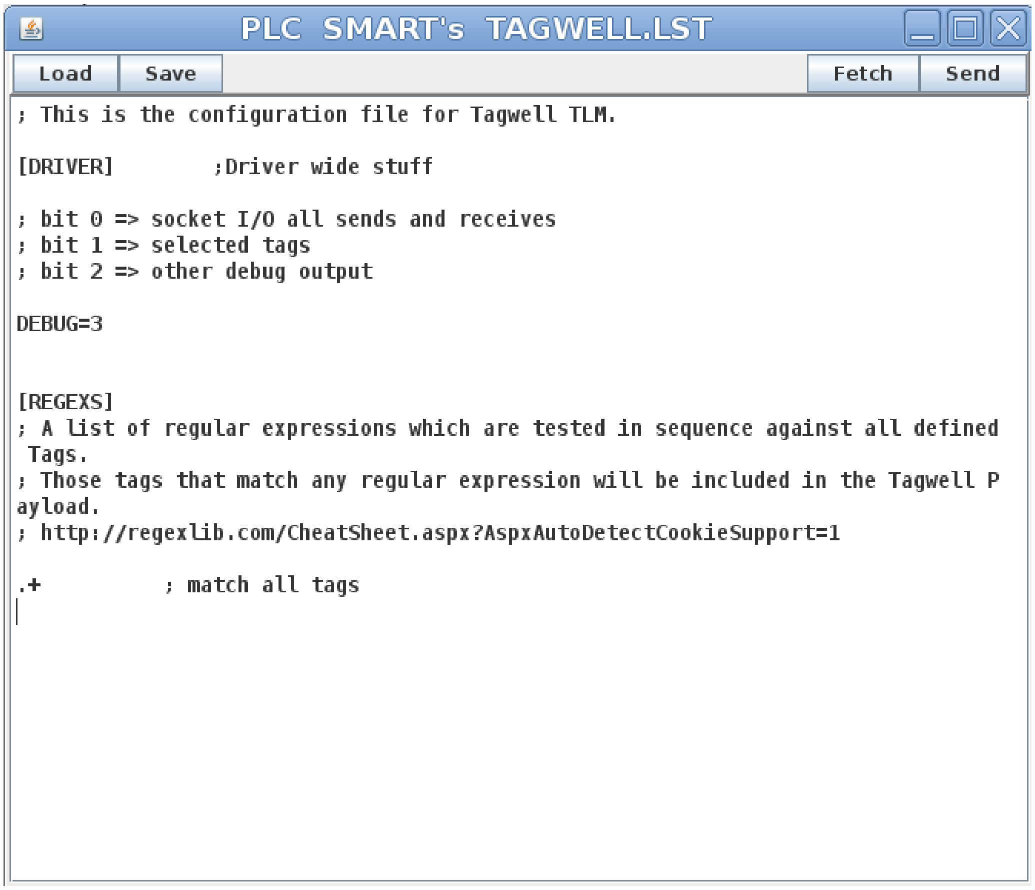
Use	Type	Name	Options
☑	DRIVER	TAGWELL.TLM	
□	DRIVER	APC.TLM	COMPORT=2 TIMEOUT=10
□	MODULE	COMGENIE.TLM	PORT0=COM2(9600,N,8,1) TIMEOUT=5 STRING...
□	MODULE	COMGENIUS.TLM	
□	MODULE	COMM.DLL	COM3IRQ=5 COM4IRQ=2
□	DRIVER	DAS100X.TLM	ADDR=00 ID=1
□	DRIVER	DIO.TLM	BASE=300 ADDR=00 BITS=96
□	MODULE	DTDISK.TLM	
□	DRIVER	ENRON_MODTC...	
□	DRIVER	GE9030.TLM	CFGFILE=/SoftPLC/tlm/GE.LST
□	DRIVER	HILSCHER.TLM	
□	MODULE	ICLC.TLM	CARD0=DC00
□	MODULE	ICO7000.TLM	COMPORT=2 BAUD=1152 TIMEOUT=20 CHECKS...

Purpose: TLI to Send a Tagwell Data Message.

Full Path: /SoftPLC/tlm/TAGWELL.TLM

Your notes on this Module

Editor Usage:



```
; This is the configuration file for Tagwell TLM.

[DRIVER]        ;Driver wide stuff

; bit 0 => socket I/O all sends and receives
; bit 1 => selected tags
; bit 2 => other debug output

DEBUG=3


[REGEXS]
; A list of regular expressions which are tested in sequence against all defined
 Tags.
; Those tags that match any regular expression will be included in the Tagwell P
ayload.
; http://regexlib.com/CheatSheet.aspx?AspxAutoDetectCookieSupport=1

.+          ; match all tags
```

Load button will load the configuration file from the development system's disk.

Save button will write the configuration file to the development system's disk.

Fetch button will load the configuration file from the runtime system's disk.

Send button will write the configuration file to the runtime system's disk.

After you edit the configuration file, the next step is to restart or cycle power on the SoftPLC in order for the changes to take effect.

> **Note:** As an alternative to cycling power, you may enter "Remote Program" mode using NexGen, then select "Remote Program" a second time. This pseudo transition from Remote Program to Remote Program is a signal to the TLM that it should reload its configuration file. This way you can load the new configuration without cycling power, although it does require you enter "Remote Program" mode (twice!).

Ladder Instruction:

TAGWELL SEND:

This instruction is used to send a data message containing the current values of the defined tag dataset. The instruction is edge triggered and will send a message when it is not busy. A total of three (3) attempts will be made to send the message.

The instruction has three (3) parameters:

"Control:" Control Element (R) address. When the rung goes TRUE, a message will be queued and the Enable (EN) and Enable Unload (EU) bits will come ON. Upon completion, the "EN" and "EU" bits will go OFF and either the Done (DN) or Error (ER) bit will come ON. If the ER bit comes ON, the Control Elements Position word (.POS) will contain an error code. When the rung is FALSE and the EU bit is OFF, the control bits are cleared (.CTL = 0).

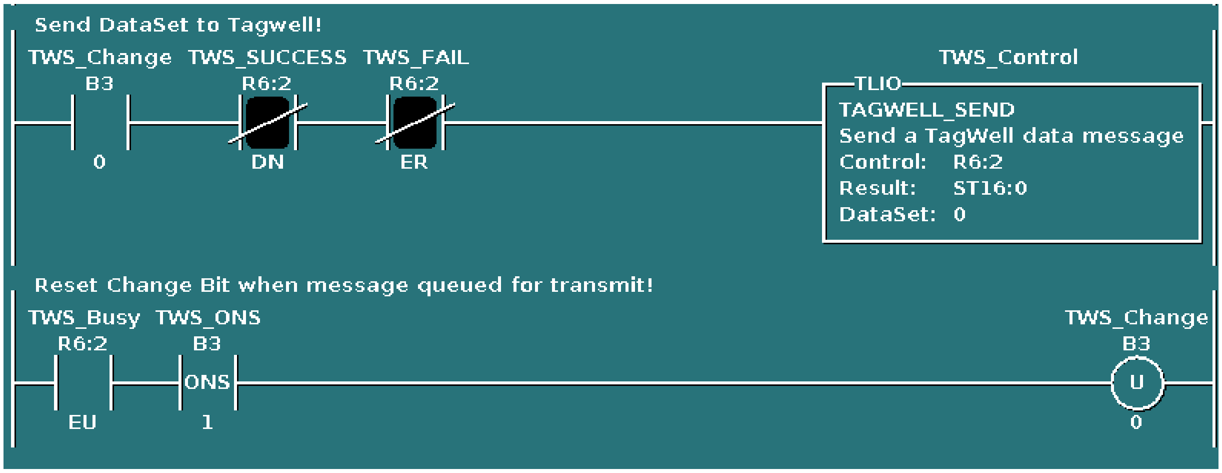
Upon completion, the Control Element ".POS" will be set to one of the following values:

0	= Success
1	= Timeout
2	= No Memory
3	= Bad Message Type
4	= Old Request
5	= Missing File
6	= Bad Config File
7	= Undefined Tag
8	= Missing Tag
9	= Invalid PLC Address
10	= Nonexistent PLC Address
11	= Bad XML
12	= Bad XML Text

"Result:" String element (ST) that will return a text string indicating success or failure. When a message is sent successfully: "Send to TagWell succeeded.". When a message send fails: "Send to TagWell failed."

"DataSet:" not implemented at this time.

Example:
When the change bit (TWS_Change=B3/0) is set to 1, the TAGWELL_SEND TLI will queue a message and set the EU bit (TWS_Busy=R6:2/EU). On the rising edge of the EU bit, the change bit is reset to zero (0) (2nd rung). Upon message success (DN) or failure (ER), the TLI's rung state will go false, the control bits will be cleared and the instruction will be ready to send another message if or when the change bit is ON.


Send DataSet to Tagwell!

TLIO
TAGWELL_SEND
Send a TagWell data message
Control: R6:2
Result: ST16:0
DataSet: 0

Reset Change Bit when message queued for transmit!

Configuration:

TAGWELL.LST Configuration file:

The configuration file for the TAGWELL TLM is **/SoftPLC/tlm/TAGWELL.LST**. This file is used to set the debug level and provide a list of tags to be shared. Tagnames may be defined using regular expressions. See TOPDOC NexGen's "Regular Expressions" Help Topic for a summary of legal regular-expression constructs. The default entry in TAGWELL.LST is:

```
.+          ;matches all defined tags.
```

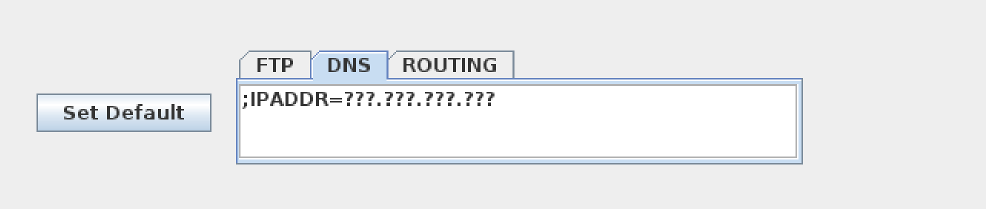
For example:

```
TWD_.+   ; would include all tagnames that begin with the characters "TWD_".
```

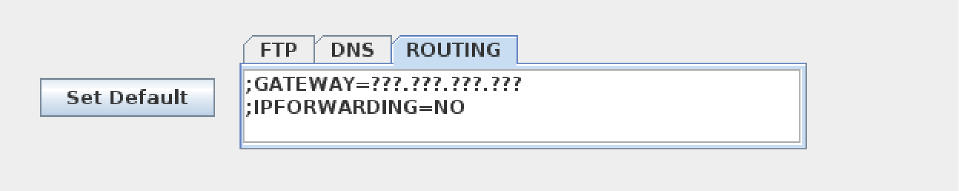
NETWORK.LST Configuration

If you are using the Tagwell Portal over the cellular VPN, you also need to do the following:

1. The wired ethernet network's DNS and Routing GATEWAY setting must not be assigned. In the PLC's Network tab, the option box should should show the following:



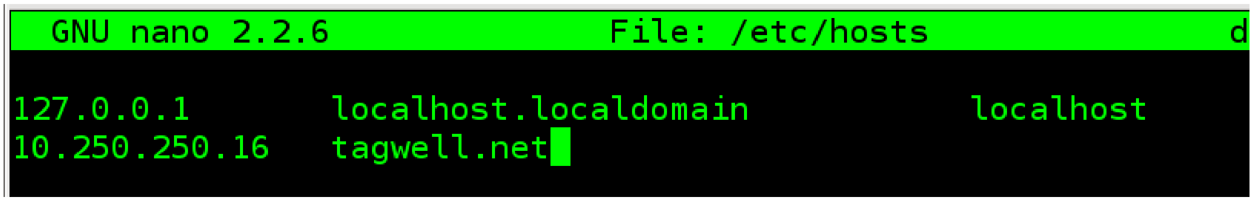
```
;IPADDR=???.???.???.???
```



```
;GATEWAY=???.???.???.???
;IPFORWARDING=NO
```

2. The Tagwell domain's private IP address must also be defined in the "hosts" file on the PLC. (/etc/hosts). From the Gatecraft Linux console, you can use the built-in text editor "nano" as follows (replace SMART with your PLC name):

```
root@SMART[/home/root]# nano /etc/hosts
```



```
  GNU nano 2.2.6                File: /etc/hosts                   d

127.0.0.1       localhost.localdomain           localhost
10.250.250.16   tagwell.net
```

Example Ladder Program:

This ladder program example can be used to send a message to TagWell when any of the defined Tags in the list changes state or value.

For this example, the following tags are shared with the Tagwell Portal.

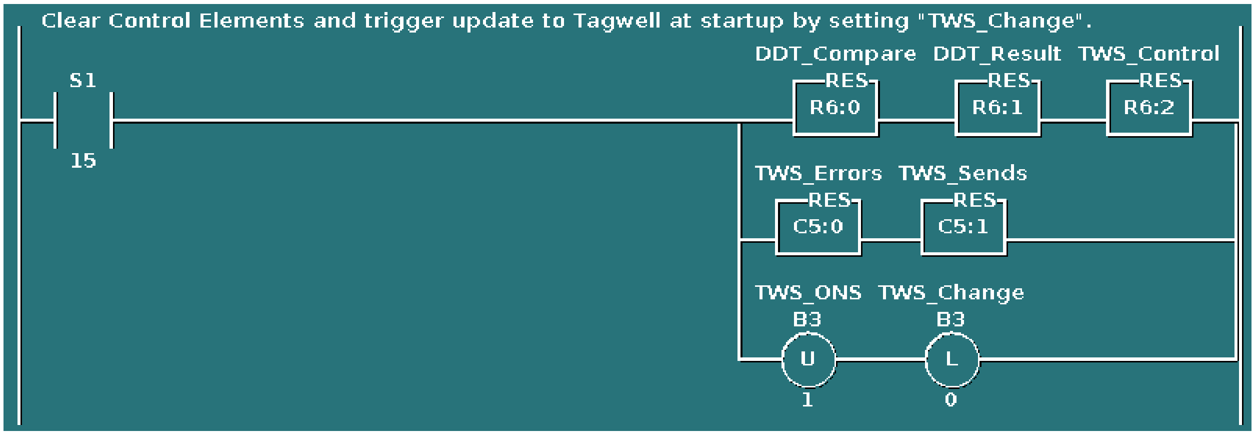
B0013/000000	TWD_Open
B0013/000001	TWD_Pump
B0013/000002	TWD_Fan
B0013/000031	TWD_Alarm
N0027:0000	TWD_AValue

Simply specifying "TWD_.+" in the TAGWELL.LST file will match all of the above tags. In TAGWELL.LST, enter the following:

```
[REGEXS]
;.+          ; match all tags
TWD_.+
```

1. When SoftPLC first starts or transitions from PROGRAM to RUN mode, the First Scan bit in the Status File (S1/15) is TRUE for one scan only. You can use it to reset all Control Elements (R*.*) and Counters (C*:*).

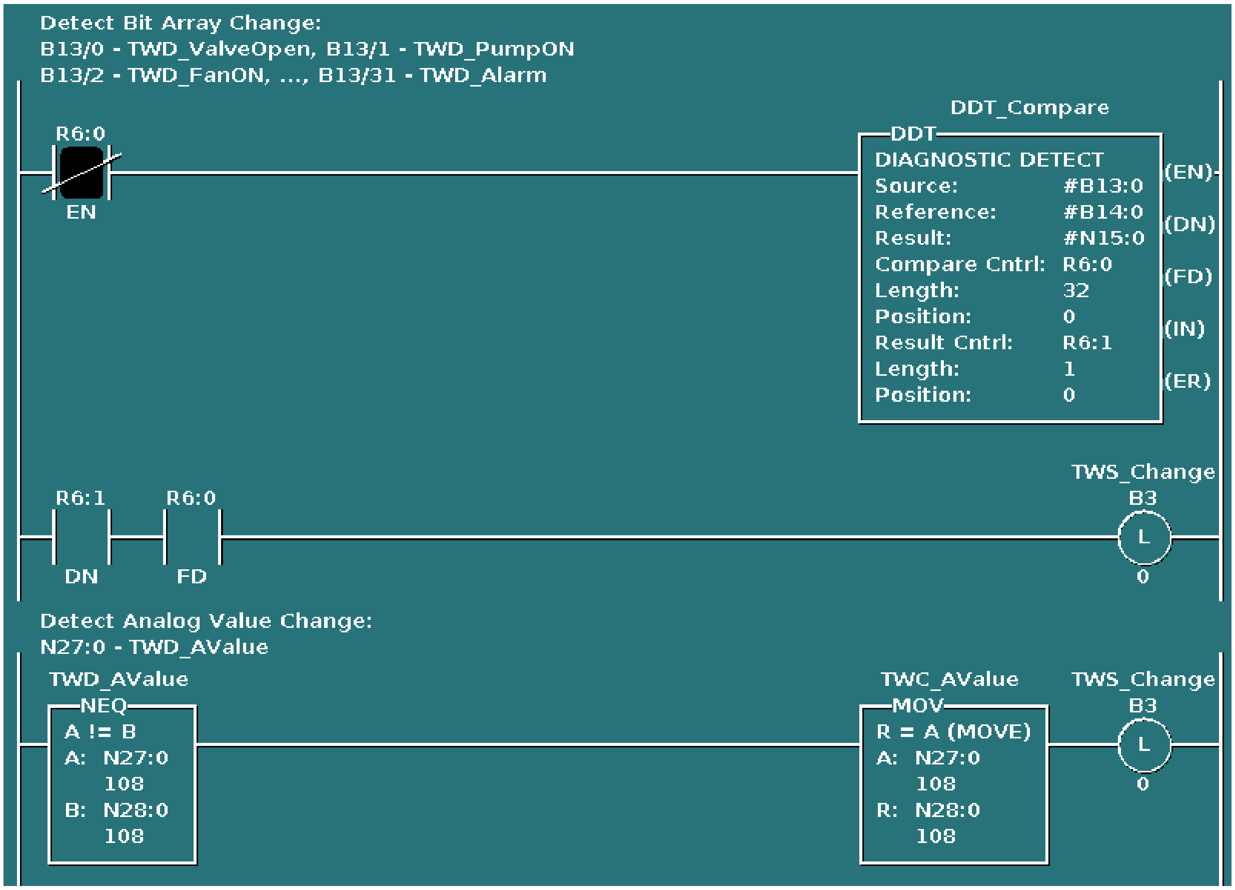
   In this example, the data change bit (TWS_Change) is also set which will initialize the Tagwell Portal.



Clear Control Elements and trigger update to Tagwell at startup by setting "TWS_Change".

2. The next three (3) rungs contain the logic to determine if any Tags shared with the Tagwell Portal have changed.

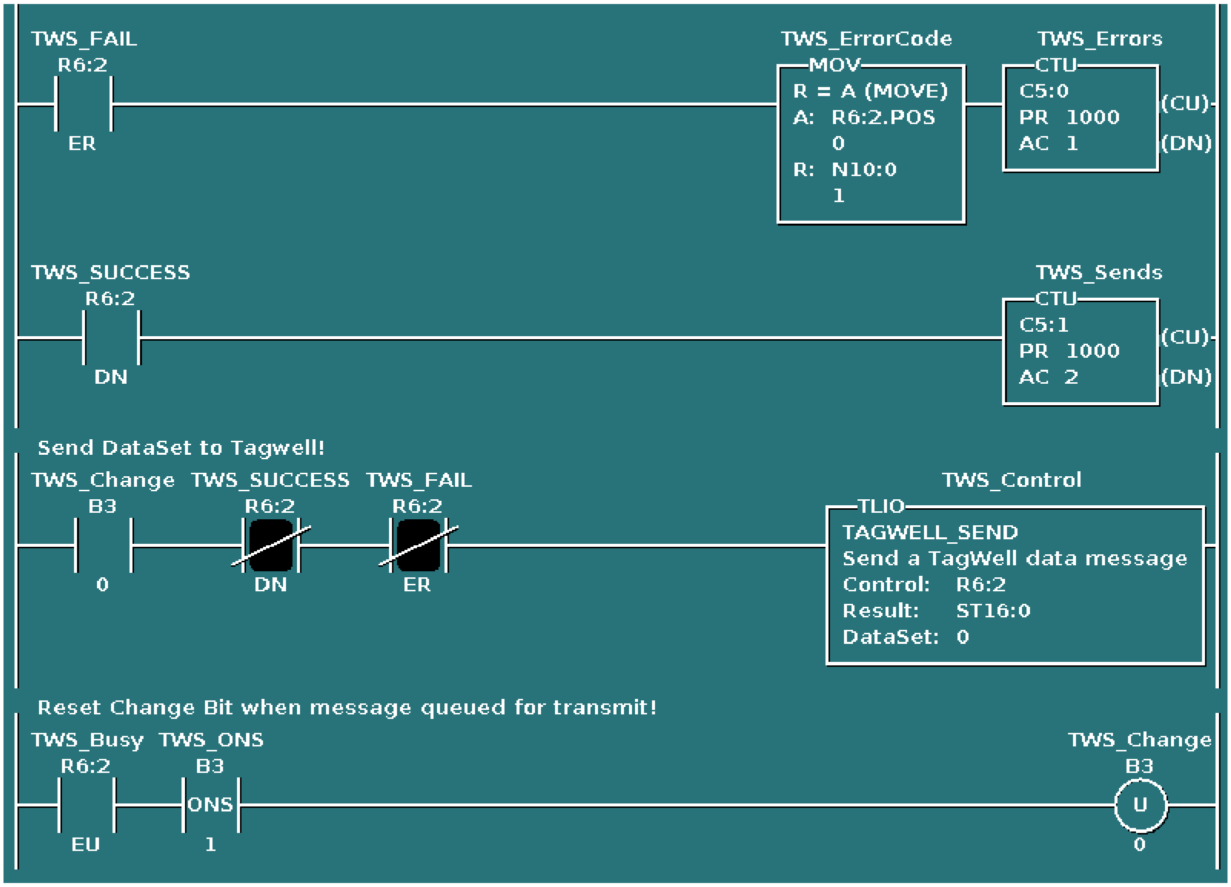
   In this example, the DDT instruction is used to detect whether any Bit Tags have changed and a "NEQ" test is used to determine if any Word Tags have changed. If any Tags are determined to have changed, latch the change bit (TWS_Change).



Note: You could use other tests to trigger a message if a value goes outside of high or low limits, or changes by a certain amount. Similarly, you could use other logic to send messages based on time of day, etc.

3. These rungs send the messages, and check for TagWell message errors.

   If a Tag has changed, send a TagWell data message.. When a new request is queued, the EU bit will turn ON and the change bit (TWS_Change) will be reset. The DN and ER counters should be increment before the TAGWELL_SEND TLI, as the Control Element's Control Word (.CTL) will be cleared when the rung is FALSE and the EU bit is OFF.


TWS_FAIL
R6:2
ER
TWS_ErrorCode
MOV
R = A (MOVE)
A: R6:2.POS
0
R: N10:0
1
TWS_Errors
CTU
C5:0
PR 1000
AC 1
(CU)
(DN)
TWS_SUCCESS
R6:2
DN
TWS_Sends
CTU
C5:1
PR 1000
AC 2
(CU)
(DN)
Send DataSet to Tagwell!
TWS_Change
B3
0
TWS_SUCCESS
R6:2
DN
TWS_FAIL
R6:2
ER
TWS_Control
TLIO
TAGWELL_SEND
Send a TagWell data message
Control: R6:2
Result: ST16:0
DataSet: 0
Reset Change Bit when message queued for transmit!
TWS_Busy
R6:2
EU
TWS_ONS
B3
ONS
1
TWS_Change
B3
U
0