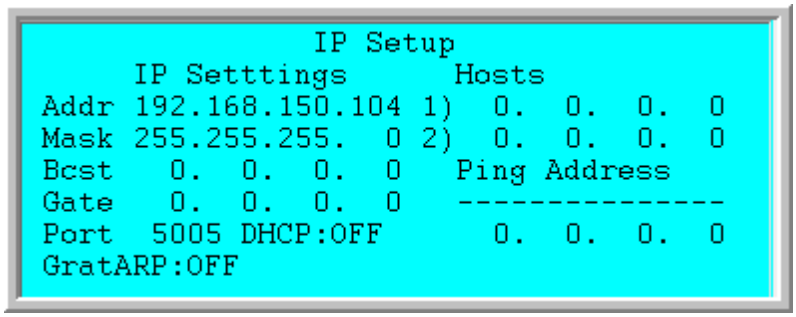
Tech Note 1114- Peer to Peer Network Setup Considerations

This TecNote reviews the considerations that are needed for Peer to Peer setups using V76.x, V80.x or Scout software. Peer to Peer uses the IO Logic screen (MM->1->8->7 or MM_>1->9->2) in each controller to allow a peer controller modify input or Output functions.

Setting Up V76.x or V80.x Controllers for Peer to Peer Communications

Typical IP Communications Setup (MM->6->5) for each Peer controller

The *IP Setup* menu configures the IP (Internet Protocol) ports implemented through the controller's Ethernet interface. The IP settings are used to identify an ATC residing on a TCP/IP network like the Station ID is used to identify a controller residing on a serial data link. The communications protocol that is used is known as UDP (User Datagram Protocol).



Most agencies typically do not use DHCP communications, so they need to set up the IP address of each local controller as per their network administrator. On this screen you will typically provide separate IP address (***Addr***) and ***Mask*** settings for the *Device* (local controller). A *Host* (central system) address can also be set up but is not necessary for Peer to Peer programming. In fact Trafficware recommends that Peer to Peer programming (MM-1->9->3) will work if the user DOES NOT program any Host IP address under MM->6->5 for communication setups that do not use DHCP. The ***Bcast*** (Broadcast) address and ***GtWay*** (Gateway) address settings are optional, but may be required for your network configuration. You must also provide an IP ***Port*** number which will match the port # in the particular drop that you are communicating with as specified. Ask your network administrator or the one who configured your network to explain how these additional settings are used if you need additional information.

The *IP Address* and *Mask* must be configured correctly for the local network. IP 1 is assigned to the local controller. The *Broadcast* and Gateway addresses can usually be set to 0.0.0.0 unless subnet addressing or routing is called for. Changes to *IP Setup* should take effect when the user leaves menu MM->6->5. As noted above, depending on the controller hardware platform, any time that you change the IP settings from menu MM->6->5, you may have to toggle controller power to cause changes in the IP settings to take effect.

DHCP (Dynamic Host Configuration Protocol) can be turned on if the agency requires it. In this case **do not** program the IP address of the local unit because one will be provided automatically by **DHCP**. In addition the user **must** program the Host ID of the central Server. This should not have any effect on the Peer to Peer data transfer.

Note: DHCP should be avoided if the agency wants to use Peer to Peer consistently. Users that use **DHCP** addressing will have to periodically go into the Peer to peer entry screen and modify the peer addressing because each peers IP address may dynamically change over time.

Peer to Peer Setup (MM->1->9->3) Screen for each Controller

The Peer to Peer programming is a way to have one controller's inputs or outputs drive another controller's inputs or outputs. It is used in conjunction with IO logic programming. Peer to Peer programming can be accomplished using any Ethernet IP connection via the programming screen shown below.

Peer	IPAddress	Port	Freq
1	192.168.104.111	5111	1.0
2	192.168.104.112	5112	2.0
3	0. 0. 0. 0	0	0.0
4	0. 0. 0. 0	0	0.0
5	0. 0. 0. 0	0	0.0
6	0. 0. 0. 0	0	0.0
7	0. 0. 0. 0	0	0.0
8	0. 0. 0. 0	0	0.0
9	0. 0. 0. 0	0	0.0
10	0. 0. 0. 0	0	0.0
11 +	0. 0. 0. 0	0	0.0

Peer: This is the Peer number assigned by the user and is programmed as ***Src*** on the IO Logic screen. The user can assign up to 15 Peers to any controller.

IPAddress: This is the Ethernet IP address of the assigned Peer controller.

Port: This is the Port number of the assigned Peer controller.

Freq:

V76.x : This is how often the Peer will be polled for information. It is programmed in seconds. Valid vales are 0-25 seconds. Typically, agencies use 1.0 for second by second polling.

V80.x or Scout: This is how often the Peer will be polled for information. It is programmed in tenths of seconds. Valid vales are 0-25.5 seconds. Typically, agencies use 1.0 for second by second polling.

Note: Peer programming can be done using up to 15 peer controllers and is two-way communications.

IO Parameters (MM->1->8->6) or (MM->1->9->1)

The Peer to Peer timeout value is programmed via the TS2 *IO Parameter* screen.

```
            I/O Parameters

C1-C11-ABC IO Mode: AUTO
    D Conn Mapping: NONE
       T&F Biu Map: DEFAULT
Invert Rail Inputs: OFF
   EVP Ped Confirm: OFF
 Peer-Peer Timeout:   0
       SIU/CMU Map: NONE
  Default Dark Map: 4
    Flash Dark Map: 4
```

Each of the possible fifteen peers that are allowed to communicate try to do so. If communications fails, this parameter will insure that I/O is not overridden by the Peer units until communications is restored. In addition this timer has the ability keep or override the peer generated input or output. If you do not get a response from the peer within the "peer to peer timeout" time, then the inputs / output for that peer all default to an **Off (FALSE)** state.

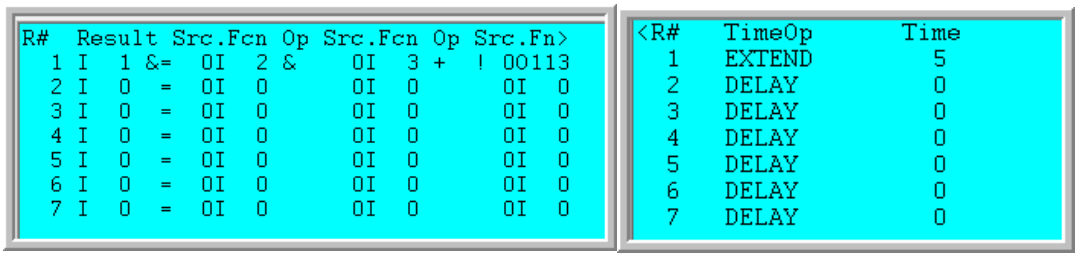
NOTE: If you program that timer as zero seconds, then the inputs/outputs from that device remain in their last known state.

Peer to Peer Comm Status (MM→1→8→8→4 or MM→1→9→7→4)

The communications status of each peer can be viewed via this screen selection. Each of the possible fifteen peers that are allowed to communicate will display the Transmit and receive block count along with any missing blocks. In addition, a Timeout value will be displayed and reset to zero each time the peer message is being transmitted and received. This will insure that each peer is actually communicating within the frequency that was programmed as per the section above.

```
     Tx.... Rx.....................
Peer Count  Count  Missed   TimeOut
  1      0      0       0       0.0
  2      0      0       0       0.0
  3      0      0       0       0.0
  4      0      0       0       0.0
  5      0      0       0       0.0
  6      0      0     + 0       0.0
```

IO Logic Basic Considerations



R#	Result	Src.Fcn	Op	Src.Fcn	Op	Src.Fn>
1	I 1	&= OI 2	&	OI 3	+	! OO113
2	I 0	= OI 0		OI 0		OI 0
3	I 0	= OI 0		OI 0		OI 0
4	I 0	= OI 0		OI 0		OI 0
5	I 0	= OI 0		OI 0		OI 0
6	I 0	= OI 0		OI 0		OI 0
7	I 0	= OI 0		OI 0		OI 0

<R#	TimeOp	Time
1	EXTEND	5
2	DELAY	0
3	DELAY	0
4	DELAY	0
5	DELAY	0
6	DELAY	0
7	DELAY	0

The *IO Logic* feature allows the user to "logically" combine IO to create new inputs and outputs that extend the functionality of the controller. On Linux platforms, up to 100 lines of Logic programming is available to the user.
The following are descriptions of each field

R#
This is the logic **R**ecord (Line) number.

Result Value and Resulting Statement
The user sets the **Result** value to either an **I** (for Input) or **O** (for Output). This selection determines if you are assigning the result of the statement to an input or an output.

Normally the resulting statement (**Result** value) equals (**=**) the logic statement that the user creates. However, with this version there is a feature where the user can also set the final **Result** value to be:

&=	Equal to the *Result value* **AND** the Logic on the right	**!&=**	Not equal to the *Result value* **AND** the Logic on the right
+=	Equal to the *Result value* **OR** the Logic on the right	**!+=**	Not equal to the *Result value* **OR** the Logic on the right
x=	Equal to the *Result value* **XOR** the Logic on the right	**!x=**	Not equal to the *Result value* **XOR** the Logic on the right

Src
This is the source controller number that is generating the logic function. The source ID will match the ***Peer*** ID number programmed on the "Peer tp Peer" menu under MM→1→9→3. Valid Source ID numbers are 0-15. Only program "**0**" as the source ID when the logic function remains within the same controller or when "Peer to Peer" programming is not used.

Fcn

This is the IO Function Number as described in Chapter 14 of the NTCIP Controller Training Manual.

The software utilizes 20 Logic Function variables numbered 230-249, where Functions 230-249 are functions "Logic 1" - "Logic 20".In addition output Logic Functions 21-30 are available and are function numbers 332-341. Whether they are denoted as input or output, they point to the same location. Think of these functions as temporary storage locations. If you want to feed the output of one statement into the next, you can make an assignment of the first statement to one of these logic variables, and then use it as a term in the next statement.

The user can optionally set a **!** prior to the **I** or **O** function. The exclamation point indicates that the term is inverted during evaluation of the statement.

Operator

This is the Logical Operation (Boolean Logic) displayed in symbols. Among the choices are: **&** (AND), **!&** (NAND), **+** (OR), **!+** (NOR), **x** (XOR), **!x** (XNOR)

The logic will follow the following truth tables-- Where '0' represents OFF or False and "1" represents ON or True

& (AND)

0	0	0		0	0	1
0	1	0		0	1	1
1	0	0		1	0	1
1	1	1		1	1	0

!& (NAND)

+ (OR)

0	0	0		0	0	1
0	1	1		0	1	0
1	0	1		1	0	0
1	1	1		1	1	0

!+ (NOR)

x (XOR)

0	0	0		0	0	1
0	1	1		0	1	0
1	0	1		1	0	0
1	1	0		1	1	1

!x (XNOR)

Timer

The timer can optionally be specified to SHIFT, DELAY, or EXTEND the result of the logic statement for the number of seconds specified by the user.

SHIFT - Shift logic by the programmed number of seconds (0-255)

DELAY - Delay logic by the programmed number of seconds (0-255)

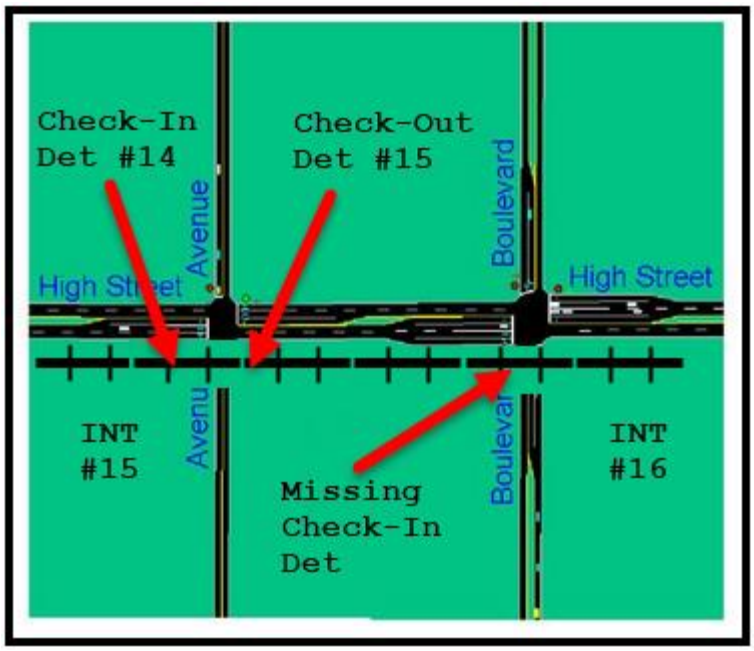
EXTEND – Extend logic by the programmed number of seconds (0-255)

SMALL DELAY – Delay logic by the programmed number of tenths seconds (0.0-25.5)

The Small Delay timer operates similar to detection delay and extend and is **ONLY** available in V80.x and Scout.

Peer to Peer Communications Example

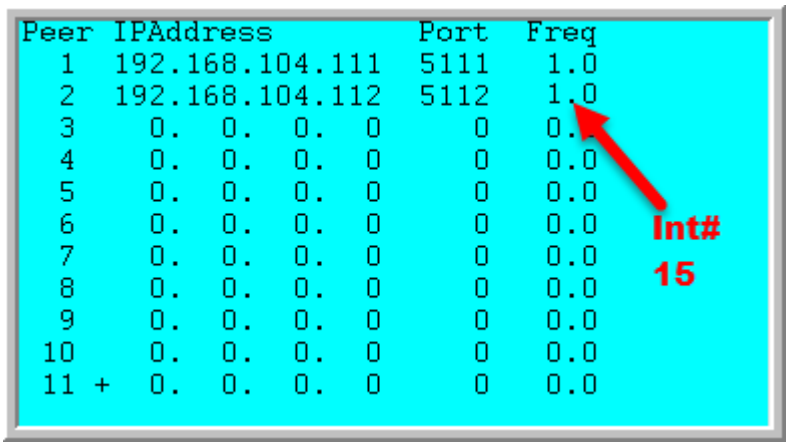
As an example, the agency has 2 intersections that are on the same corridor that utilize Light Rail TSP Check-in and Check-Out detectors.



Intersection # 15 has a Check-Out detector on the LRT track that is associated with Det # 15 (Input function 15). Intersection #16 is missing a Check-in detector. We can use Peer to Peer and Logic to have Detector #15 from Intersection 15 drive a "dummy" Check-in detector for intersection #16 (we will use detector # 35). The screens below are programmed on the Intersection #16 controller.

Peer Setup (MM->1->9->3)

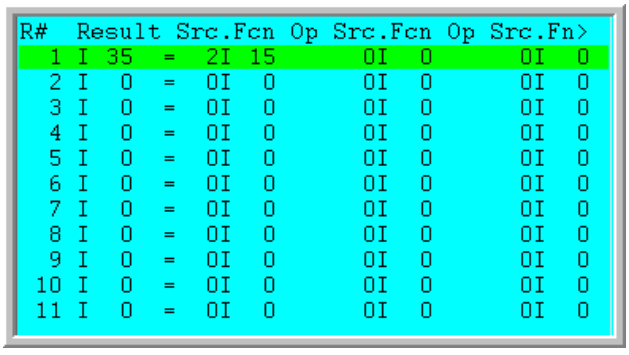
First we need to define the Peer relationship in Intersection #16. For this example Intersection #16 has 2 peers Intersection #14 and Intersection #15. Intersection #14 uses IP address 192.168.104.111 via Port 5111 and Intersection #15 uses IP address 192.168.104.112 via Port 5112. We will poll the peer one a second. The programming for this setup is shown below via **MM->1->9->3.**

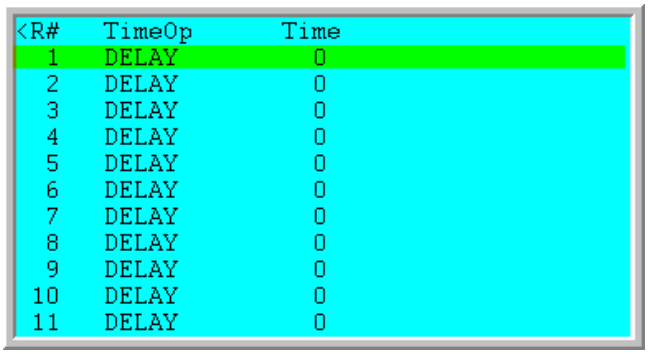


Note that Peer #1 is Intersection #14 and Peer #2 is Intersection 15.

Logic Setup (MM-1-8-7)

Next we will program the Intersection # 16 logic screen (**MM->1->8->7**) to use Intersection #15 detector # 15 as the source for Detector #35. Intersection #15 is peer #2 so we will use "2" as the source as shown below.





Summary

By following this procedure, you will be able to will be able to remotely connect, view and change any parameter of the controller database subject to the Agencies security protocol and policies.