# TecNote 1204 – Preemption Service Delay (PSD) setup using Transit Preemption Detectors and the Transit Preemption Matrix


Preempt Service Delay (**PSD**) eliminates most of the inefficiencies at light rail intersections that occur when using preemption. **PSD** is particularly effective when it is used combined with priority hold phases and dynamic exit.

**PSD** was developed for use at intersections where the Light Rail Transit (**LRT**) movement is controlled by the traffic signal controller and where **LRT** has priority over other traffic. The City of Houston has mandated that any preemption for **LRT** does not allow a reduction of pedestrian walk or Ped Clearance times. As a result, the ability for the controller to get to the dwell phase is unpredictable and can cause delays for **LRT** vehicle. The delay is caused by the pedestrian walk and Ped Clr times needing to be completed. Additionally, the preempt input might come in when the there is no ped in service and this may cause an unnecessary truncation of the vehicle movements at the intersection. This becomes very noticeable in a central business district where all the pedestrian signals are on recall.

The delay caused by the pedestrian Walk + Ped Clr + Yellow + Red is referred to (in this document) as Pedestrian Yield (**PY**) time. To compensate for **PY** delay, the preemption input is turned on earlier than **LRT** service is needed. This allows service even if the worst-case delay is encountered. However, it is very unlikely that an intersection is going to be delayed by the entire **PY** time and in most instances the preempt will start earlier than needed, causing users of the intersection to be delayed. Also, since a preempt can start up to **PY** early, the maximum duration that the preempt is allowed to run must be increased by **PY** time to ensure its effectiveness.

Consider two adjacent intersections. The first intersection started a conflicting pedestrian move immediately prior to receiving an **LRT** preempt input. The second intersection is at a portion of the cycle where it can respond immediately to the **LRT** preempt input. The second intersection will dwell for **PY** plus the travel time prior to the light rail vehicle arriving at the intersection. While this functions to keep the light rail vehicle moving, it is very inefficient and frequently results in citizen complaints. In heavy traffic conditions the extra delay at the second intersection can cause spill back at other adjacent intersections.

**PSD** provides the same level of service to the light rail vehicle as traditional preemption described above without the inefficiency of traditional preemption inputs. **PSD** keeps each intersection cycling during **PY** time and each phase or pedestrian signal that could delay the start of the preempt is inhibited at the point it would impact the start of the preempt. The preempt is applied only at the point that the preempt phase will start running at **PY** time.

Preemption service must also consider the situation where the light rail train does not arrive in a reasonable amount of time. If the vehicle for which priority is being provided fails to use the priority in a reasonable amount of time, the intersection should resume cycling for other users. The value of "reasonable amount of time" is a discretionary consideration and will need to be set for each system. If the input does not drop (indicating the train has cleared the intersection) within the "reasonable amount of time" it is considered to have timed-out/failed. Each preempt input has a user definable time-out value. Once an **LRT** input has timed-out, and no other non-timed-out **LRT** inputs are on, the intersection will run the time-out preempt. Note there can be instances where the first **LRT** input is timed out, but a second **LRT** input started prior to the first timing-out. In this instance the controller will time the preempt for the second input prior to entering the time-out preempt.

The time-out preempt returns the intersection to cycling. All phases and overlaps are enabled in the time-out preempt. The intersection runs free and uses the values from **MM->1->1->1**. The time-out preempt remains running until all **LRT** inputs drop. The City of Houston normally sets relatively short min and max green times for time-out, so the intersection will cycle quickly. Typical starting values are 10 seconds max green for **LRT** phases, 15 seconds max green for left turn and side street phases, and 25 seconds max green for major street phases. One key practice is to set the min green for **LRT** phases to zero, this prevents conflicting outputs when a train clears the intersection during time-out and the controller exits preemption. During fail-safe preempt operation no coordination is running.

Two additional features are required to get the best use from **PSD:** priority hold phases and dynamic exit. Dynamic exit instructs the traffic controller to service the phase that has been waiting the longest at the end of a preempt. Dynamic exit is available to all preempts in Trafficware V80.x and V85.x/Scout software and is not specific to **PSD** operations. Priority hold phases instruct the controller to remain in a flagged phase until the start of the **PSD** preempt if the controller is currently in that phase. Priority hold phases are used at locations were the majority of traffic is on approaches that conflict with the light rail approaches. Defining priority hold phases allows a corridor to remain green until forced off by the **PSD** preempt. works by eliminating most of the inefficiencies at light rail intersections that occur when using preemption. **PSD** is particularly effective when it uses combining priority hold phases and dynamic exit.

**PSD** was developed for use at intersections where the Light Rail Transit (**LRT**) movement is controlled by the traffic signal controller and where **LRT** has priority over other traffic. The City of Houston has mandated that any preemption for **LRT** does not allow a reduction of pedestrian walk or Ped Clearance times. As a result, the ability for the controller to get to the dwell phase is unpredictable and can cause delays for LRT vehicle. The delay is caused by the pedestrian walk and Ped Clr times needing to be completed. Additionally, the preempt input might come in when the there is no ped in service and this may cause an unnecessary truncation of the vehicle movements at the intersection. This becomes very noticeable in a central business district where all the pedestrian signals are on recall.

The delay caused by the pedestrian *Walk + Ped Clr + Yellow + Red* is referred to (in this document) as Pedestrian Yield (**PY**) time. To compensate for **PY** delay, the preemption input is turned on earlier than LRT service is needed. This allows service even if the worst-case delay is encountered. However, it is very unlikely that an intersection is going to be delayed by the entire **PY** time and in most instances the preempt will start earlier than needed, causing users of the intersection to be delayed. Also, since a preempt can start up to **PY** early, the maximum duration that the preempt is allowed to run must be increased by **PY** time to ensure its effectiveness.

## Transit Detectors and Concepts

Transit Detectors are used to for Light Rail vehicles. Detection is used for Light Rail Transit Priority to check the Train in and out. Transit Detection is programmed under the MM->5->9->8 screen. In particular:

**Advanced Detectors** – advanced detectors are special detectors that utilize Preempt Service Delay (**PSD**) operations. If a detector is programmed in this field, it will be used to call an internal preempt service delay algorithm that will automatically calculate the delay to be applied for preempt service. Internally, the controller will calculate the longest delay time based on conflicting phase and pedestrian movements. Typically, the longest delay is the conflicting pedestrian movement, calculated by adding the *Walk + Ped Clr + Yellow + Red*, but can also be calculated using the longest delay caused by a conflicting phase minimum green + yellow + red. The controller will continue to service conflicting movements until no other conflicting movements can be serviced (using the above calculation). As each movement can no longer be serviced, the controller will apply a phase/ped inhibit to prevent those movements from being serviced. This operation can be modified by programming a hold phase in the Transit Preempt Matrix and flagging the detector to USE HOLD in MM-5-9-8+. See below.

**Check in detectors (ChkInDet)** – check in detectors work as a latching input that will remain active until a check out input is received, or the max duration time expires. Note that if the controller does not receive a check out and no time is programmed in max duration, the check in will not expire.

**Check out detectors (ChkOutDet)** – check out detectors are used to release the check in detector once it is latched. Check in and check out detectors do not need to be different detectors. For example, if det19 is both the check in and check out detector, while det19 is on the detector is checked in and when det19 turns off the detector is checked out. This is beneficial when only one detector is available and no check out or downstream detector is available. Logic can be utilized to extend a pulsed detector if that is the case, simulating a check in and check out configuration.

**Max Duration (MaxDur)** – max duration is the amount of time the advanced detector or check in detector will remain active before timing out an releasing. The input is still active, but the controller will no longer respond to it until it is cycled off and then on again. Max duration is timed from the beginning of the associated preempt dwell interval

**Check in delay (ChkInDly)** – check in delay is timed from the beginning of the check in detector going active

**Lock Out Time (L/OTime)** – amount of time between activations before a new check in can be acted upon

**Preempt** – the transit detector can also directly call a preempt when it is active. Programming a preempt in this field will cause that preempt to be activated when the detector is active or latched

**Use Hold (UseHold)** – flagging this entry will cause the controller to reference the transit detector matrix to determine what phase(s) to hold when the detector is active or latched. The hold phase operation will only occur if the hold phase(s) are currently active or when they become active. As with **PSD**, the controller will attempt to service any phase that can be serviced while meeting its minimum *Green + Yellow + red* without causing the preempt to be delayed

## Transit Detector (MM->5->9->8) Programming Considerations

### Programming Version V80.5F or later and Scout V85.2 or later

Up to eight Light Rail or Transit Priority **LRV** detection selections can be programmed to check the light rail or transit vehicle in and out. **NOTE: This feature will only work if Preemption Type is set to LRV under MM->3->1->6.**

**TLRMode** is the key field to program when using this screen.

When **TLRMode** is set to **ON** (**X**) the intersection is running in the Transit Light Rail (**TLR**) Mode. In this mode, a Transit vehicle is running over specified detectors to create the preemption.

Transit Light Rail Detectors

| LRV Det | ..1 | ...2 | ...3 | ...4 | ...5 | ...6 | ...7 | ...8 |
|---|---|---|---|---|---|---|---|---|
| AdvDet | 0 | 0 | 15 | 17 | 0 | 0 | 0 | 0 |
| ChkInDet | 9 | 11 | 0 | 18 | 25 | 0 | 0 | 0 |
| ChkOutDet | 10 | 12 | 16 | 19 | 26 | 0 | 0 | 0 |
| MaxDur | 30 | 30 | 30 | 30 | 0 | 0 | 0 | 0 |
| ChkInDly | 0 | 20 | 0 | 5 | 0 | 0 | 0 | 0 |
| TLRMode | X | X | . | . | X | . | . | . |
| L/OTime | 0 | 0 | 6 | 6 | 0 | 0 | 0 | 0 |
| OvrTime | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reserved | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Preempt | 9 | 10 | 0 | 0 | 0 | 0 | 0 | 0 |
| UseHold | . | . | . | . | . | . | . | . |

When **TLRMode** is set to **OFF** (**.**) the intersection is running in the Preemption Service Delay (**PSD**) Mode. The Preemption Service Delay (**PSD**) Mode is intended for use with an input coming from external source like a rail cabinet or GPS. It is automatically calculated by the software. The user must verify (calculate) the **PSD** time and give it to the agency that is providing the **PSD** input so they can adjust as needed to reduce impact of the Transit.Time. Please note that the **PSD** time is calculated to be equal to the longest programmed **walk + pedestrian clearance + yellow clearance + red clearance times.**

**The rest of the fields on this screen are based on the TLRMode setting that the user programs.**

**AdvDet (Advanced Detector)** – This is the detector number that will place the initial call to the Transit Phase. It will initiate the TSD (Time-of-Service-Desired) counter to the Light Rail Transit Priority service phase.

**ChkInDet (Check-In Detector)** – This is the detector number that tells the controller that the train has arrived for service. This detector will place a call to the Transit Phase if there is not one existing from the Advanced Detector.

**ChkOutDet (Check-Out Detector)** – This is the detector number that tells the controller that the train has cleared the intersection.

**MaxDur (Max Duration)** – The maximum amount of time, in seconds, that the Check-In detector will apply an input before it is automatically checked out. This is to avoid "Stuck Detection" from holding the Green.

**ChkInDly (Check-In Delay)** – This parameter acts like the Preemption Delay timer in FREE mode. This is the delay time, in seconds, for the Advanced Detector Input while in FREE operation, because in FREE mode the Light Rail Vehicle (LRV) is serviced with Transit Preempt service. This parameter may be set and should be the same value as the Time of Service Desired (TSD) value in coordination.

**L/OTime (Lockout Time)** – The amount of time, in **minutes**, that must elapse between requests to be serviced for that direction. It prevents another Check in Detector input calling a preemption. The user has this option to prevent multiple preemption's coming in for the same direction back to back.

**OvrTime (Override Time)** – is a fail-safe timer for a stuck cabinet override input. It is used in association with any of Cabinet Input functions that can override the PSD Mode. **OvrTime** is a fail-safe timer if any of these cabinet override inputs is stuck ON. Refer to the following page for further information.

**Preempt –** This is the associated high priority preemption number (1-12)

**UseHold –** Use the programmed Hold Phases selected on the **MM->5->9->5** screen and hold them for the selected **LRV** preemption. These Hold phases are selected prior to the high priority preemption being run. This provides the user option to hold phases green until preemption causes phases to terminate clearing traffic prior to Transit vehicle arriving at intersection.

In addition, **Output Function #138 (LRV Warning Status Output)** can be mapped to drive any cabinet output. For instance, this output could be wired to a "Train Coming" sign. When any Transit Light Rail detector is activated, and the **TLR** or **PSD** commences, this output will come on. It will remain on until the **TSD** or **PSD** is completed.

Typical Logic programming under **MM->1->8->7** for flashing (O113) and driving Special Function Output 1 (O103) from the LRV Warning Status Output (O138) is as follows: **O 103 = O138 AND O113**

Each LRV detector depends on the above programming and the programming will utilize software to select the LRV input that will be used for high priority Preemption detection described in the **TranPreMatxDet** section.

**Operational considerations based on the TLRMode settings**

With **TLRMode** set to **ON:**

The Transit Light Rail (**TLR**) Mode is intended for use with Raw Detector inputs from the intersection. The Transit vehicle runs over detectors in the intersection bringing inputs into controller like normal vehicle inputs.

- The Check-In detector should call the preemption immediately and stay in the preemption until the Check-Out Detector input is received or until the Max Duration timer expires.
- The Advance Detector should become the Check-In Detector after the programmed Check-in Delay times expires. This input will call the preemption until Check-Out Detector input is received or the Max Duration timer expires. If the Check-in Delay timer is set to "**0**" the software will call the preemption immediately. Further note: if the Check-Out Detector **is not** programmed, check-out will occur at the trailing edge of the Advance Detector call (i.e. the Advanced Detector call is dropped).
- The L/O Time is a timer (in minutes) that user can program to prevent another Check in Detector input calling a preemption. The user has this option to prevent multiple preemption's from coming in for the same direction back to back.

With **TLRMode** set to **OFF**:

The Preemption Service Delay (**PSD**) Mode is intended for use with an input coming from external source like a rail cabinet or GPS. It is automatically calculated by the software. The user must verify (calculate) the **PSD** time and give it to the agency that is providing the **PSD** input so they can adjust as needed to reduce impact of the Transit.Time.  Please note that the **PSD** time is calculated to be equal to the longest programmed **walk + pedestrian clearance + yellow clearance + red clearance times.**

- The Advance Detector should put intersection in Free, place Inhibits on all PEDS immediately, use the **PSD** time as a delay, then calls the preemption until the Check-Out Detector input is received or Max Duration time expires.
- NOTE: The **Check-In** Detector, **Check-in Delay** and the **L/O Time** are not used under **PSD** Mode.

**Override input functions 205, 532-539** and **OvrTime** (**PSD only**):

Nine Cabinet Inputs can be assigned a function numbers to override the calculated **PSD** time described in the above section.

**Input #205**: Apply inhibit phases for all Rail Dets immediately

**Input #532**: Apply all inhibit phases for Rail Det 1 immediately

**Input #533**: Apply all inhibit phases for Rail Det 2 immediately

**Input #534**: Apply all inhibit phases for Rail Det 3 immediately

**Input #535**: Apply all inhibit phases for Rail Det 4 immediately

**Input #536**: Apply all inhibit phases for Rail Det 5 immediately

**Input #537**: Apply all inhibit phases for Rail Det 6 immediately

**Input #538**: Apply all inhibit phases for Rail Det 7 immediately

**Input #539**: Apply all inhibit phases for Rail Det 8 immediately

If the external source that the **PSD** Mode is using is not working correctly, a technician can physically turn on a cabinet input to override and turn off the **PSD** Mode calculations. This will cause the software to revert back to the **TLR** Mode. This provides the user flexibility to have programmed Check-In Delay times (which are ignored when in **PSD**). If the Override input is activated, then **OvrTime** will be used. The **OvrTime** is a fail-safe timer for a stuck cabinet override input.

This timer shall start to countdown once override input comes in and if time expires before input drops, the intersection will go to a programmed failed preemption selected on **MM->5->9->5**.

**Advance Detectors**

If advance detectors are programmed, the Transit Light Rail preemption will be delayed automatically by the longest delay (i.e. this delay accounts for the worse case movement). The worse case movement is the one that takes the longest to clear and is typically the pedestrian movement. So, if you have an advance detector go active, the controller looks to see what the longest delay is and delays the input. As the wait is timed down, inhibits are applied to phases that cannot serve anymore due to not enough time being left.

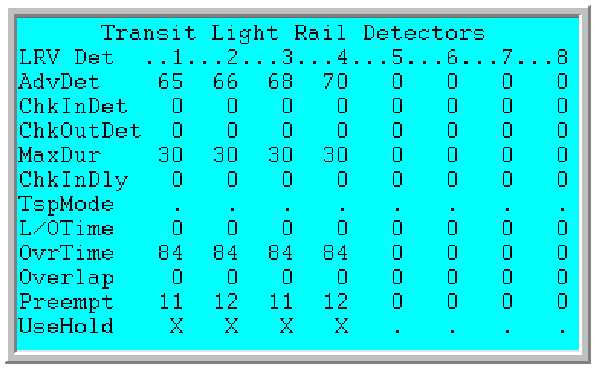
**Max Check In** – This time is based on the input being active, not necessarily the preemption being in service. This timer is different that the **Max Duration** (**MM->3->1->7**).

**Check In / Check Out detectors** - the check in and check out detectors can be the same or different detectors. If they are the same, the rising edge actuation checks it in and the falling edge actuation checks it out

## Transit Detector (MM->5->9->8) Programming Considerations

### For V80.5E or earlier and Scout V85.1 or earlier

Up to eight Light Rail or Transit Priority **LRV** detection selections can be programmed to check the light rail or transit vehicle in and out. **NOTE: This feature will only work if Preemption Type is set to LRV under MM->3->1->6.**



Transit Light Rail Detectors

| LRV Det | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| AdvDet | 65 | 66 | 68 | 70 | 0 | 0 | 0 | 0 |
| ChkInDet | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ChkOutDet | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MaxDur | 30 | 30 | 30 | 30 | 0 | 0 | 0 | 0 |
| ChkInDly | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TspMode | . | . | . | . | . | . | . | . |
| L/OTime | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| OvrTime | 84 | 84 | 84 | 84 | 0 | 0 | 0 | 0 |
| Overlap | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Preempt | 11 | 12 | 11 | 12 | 0 | 0 | 0 | 0 |
| UseHold | X | X | X | X | . | . | . | . |

**AdvDet (Advanced Detector)** – This is the detector number that will place the initial call to the Transit Phase. It will initiate the TSD (Time-of-Service-Desired) counter to the Light Rail Transit Priority service phase.

**ChkInDet (Check-In Detector)** – This is the detector number that tells the controller that the train has arrived for service. This detector will place a call to the Transit Phase if there is not one existing from the Advanced Detector.

**ChkOutDet (Check-Out Detector)** – This is the detector number that tells the controller that the train has cleared the intersection.

**MaxChkIn (Max Check-In)** – The maximum amount of time that the Check-In detector will apply an input before it is automatically checked out. This is to avoid “Stuck Detection” from holding the Green.

**ChkInDly (Check-In Delay)** – This acts like the Preemption Delay timer in FREE mode. This is the delay time for the Advanced Detector Input while in FREE operation, because in FREE mode the Light Rail Vehicle (LRV) is serviced with Transit Preempt service. If set, this parameter should be the same value as the Time of Service Desired (TSD) value in coordination.

**TspMode** – If the agency has the Transit module enabled and licensed then setting this parameter on will allow the detectors to be used during Transit Priority.

**L/OTime (Lockout Time)** – The amount of time in seconds that must elapse between requests to be serviced for that direction

**OvrTime (Override Time)** – Drives Output Function Code #138 (LRV Warning Status Output) such as a "Train Coming" sign as shown above. This value sets the amount of time in seconds that the LRV sign display is activated BEFORE the LRV is serviced. In Coordination, the output is activated prior to the end of the TSD value by the Output Time value. The output will remain active until the Check-Out detector is activated. In FREE operation the output is activated prior to the end of the "Check-In Delay" value by the Output Time value. To drive this output function a Special Function Output Load Switch channel, or equivalent, must be assigned to this function through I/O Logic.

Typical Logic programming under **MM->1->8->7** for flashing (O113) and driving Special Function Output 1 (O103) from the LRV Warning Status Output (O138) is as follows:

**O 103 = O138 AND O113**

**Overlap -** When you program an overlap number for a specific transit priority input, the software will place a call on **all** included phases in the overlap

**Preempt –** This is the associated high priority preemption number (1-12)

**UseHold –** Use the programmed Hold Phases selected on the **MM->5->9->5** screen and hold them for the selected **LRV** preemption. These Hold phases are selected prior to the high priority preemption being run.

Each **LRV** detector depends on the above programming and the programming will utilize software to select the **LRV** input that will be used for high priority Preemption detection described in the next section.

## Transit Preemption Matrix (MM->5->9->5)

```
                Pre Det.1.2.3.4.5.6.7.8
                 0   -   . . . . . . . .
                 0       . . . . . . . .
                 0       . . . . . . . .
                 0       . . . . . . . .
                 0       . . . . . . . .
                 0       . . . . . . . .
                 0       . . . . . . . .
                 0       . . . . . . . .
TimeOutPreempt  0
HoldPhases     1          2          3
    12345678 90123456 78901234 56789012
    ........ ........ ........ ........
```

In this screen, each of the Transit Detector inputs are across the top and the associated preempt is down the left side. The preempt matrix is useful when multiple transit detectors are being used to display different indications or run different operations. For example, if transit detector 1 calls NB rail and transit detector 2 calls SB rail, transit detectors 1+2 can call NB & SB. This is very useful if the application has multiple parallel tracks/paths and the vehicles can run in either direction on either path and at the same time. An example would be NB normal and NB reverse (a vehicle running NB on the SB track).

For each of the programmed entries, the controller evaluates the matrix and stops when it finds an exact match. This is very important to note as the user should program the entries in the order of importance. When using the transit matrix, priority levels in the individual preempt are overridden by the matrix, so it is very important to enter the input combinations in the order the user wishes them to be evaluated.

**Pre** – the preempt to be called when the preempt detector matrix has an exact match.

**Det 1-8** – In the Classic User Interface, each entry can be programmed by first moving the cursor over to the det field and then pressing the numbers 1-8 to toggle the flag.

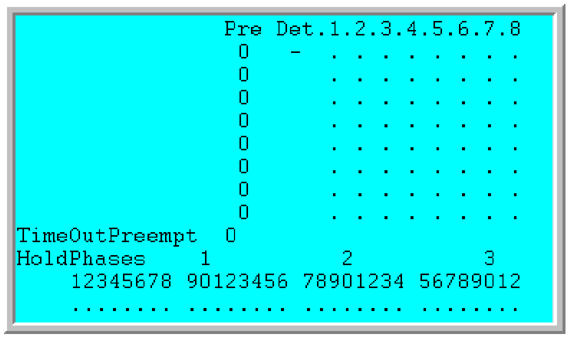
**Time Out Preempt** – a time out preempt can be programmed to run if the transit detector max duration is reached. Typical application would be where a light rail input does not go away and instead of not acting on the input anymore or remaining in preempt, the controller transitions to a preempt that services both the failed input and the vehicles/peds that have been waiting. This creates a failed state that allows the intersection to cycle somewhat normally until the failed input(s) turn off. Note that the failed state is considered active until ALL of the transit detectors turn off. Also, the controller will not enter the failed state if another detector is timing its preempt service delay.

**Hold Phases (1-32)** – programming hold phases will cause the controller to remain in the hold phase if it is already in service. For example, if the controller has been programmed to hold phases 2 & 6, when a transit detector goes active, it will stay in 2 & 6 if they are already green. If the controller is not in phases 2 & 6, it will serve any phases that meet the min green+yellow+red qualifier until it either reaches the hold phases or cannot serve any phases safely before the preempt.

## Transit Preemption Matrix example

### Version V80.5F or later and Scout V85.2 or later

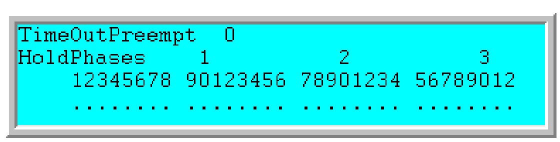
This matrix allows the user to select a particular LRV detection selection (**MM->5->9->8**) that may be used for each preemption. Each of the first twelve rows row represents a user programmed High Priority Preemption (1-12) and each column represents the **LRV** detection selected for that preemption. See the above section for a discussion on the programming of the **LRV** detection.




**NOTE: This feature will only work if Preemption Type is set to LRV under MM->3->1->6.**

**Pre**: The user will program the High priority Preemption (1-12). A value of "**0**" will disable the row,

**Det:** The user can select any combination of LRV detection (columns 1-8). The selected columns are used by the software to evaluate if the particular programmed Transit detector **(MM->5->9->8**) will be used based upon the preemption priority hierarchy. This matrix is subject to the preemption priority hierarchy programmed under **MM->3->3 and MM->3->8 34** described earlier in chapter 8.




**TimeOutPreempt (0-12)** – The user can choose a fail-safe preemption (1-12) that will be run if the LRV detector check-in timer has expired. This check-in timer is calculated by the specific user program settings. Setting **TimeOutPreempt** to 0 will remain in the selected High priority preemption chosen via the matrix programed above.

**Hold Phases** - Any Phases (1-32) can be selected as a Hold Phase. Hold phases are phases that are held till the selected preemption, via the above matrix, becomes active. These phases will be held if the **UseHold** Parameter under **MM_>5->9->8** is set for the specific LRV preemption. If there is a hold phase programmed, and if that phase is active, the controller will not process any other movement until the LRV preempt longest delay time expires. The purpose of this is to keep the controller in the phase that is currently holding if it is already in service.  However, if the Hold Phase is not in service, the intersection will continue to cycle until it reaches the hold phase or the preempt delay timer expires.

The **TranPreMtrxDet** screen is used when the agency requires combinatorial logic to call a preemption, using a programmed matrix. The purpose that this was specified and created was to save entering data in the matrix and slots for programming. This is also why the preemption is able to be specified in the preempt matrix. For example, the Transit Light Rail Detector # 1 could call preempt 1, but entry 1 in this matrix does not need to reference preempt 1. The software algorithm uses the first line of matching logic to see what preemption it will run with the combination of inputs.

The following example will discuss the programming of both the TranDet (**MM->5->9->8**) and TranPreMtrxDet (**MM->5->9->5**) screens.

As an example, the user has programed the standard preemption hierarchy where HP1 overrides HP2, overrides HP3, etc. Below are screens for **MM->5->9->8**, the Transit Check-in detectors, and **MM->5->9->5** the Transit Preemption detector Matrix:

```
      Transit Light Rail Detectors
LRV Det   ..1...2...3...4...5...6...7...8
AdvDet      0   0  13  15   0   0   0   0
ChkInDet    9  11   0  16  18   0   0   0
ChkOutDet  10  12  14  17  19   0   0   0
MaxDur     30  30  30  30  30   0   0   0
ChkInDly    0   0   0   0   0   0   0   0
TLRMode     X   X   .   .   X   X   X   X
L/OTime     0   3   0   0   0   0   0   0
OvrTime     0   0   0   0  30   0   0   0
Reserved    0   0   0   0   0   0   0   0
Preempt     9  10   3   4   5   0   0   0
UseHold     .   .   .   .   X   .   .   .
```

```
Pre Det.1.2.3.4.5.6.7.8
 7      X X . . . . . .
 8      . . X X . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0      . . . . . . . .
 0   +  . . . . . . . .
```

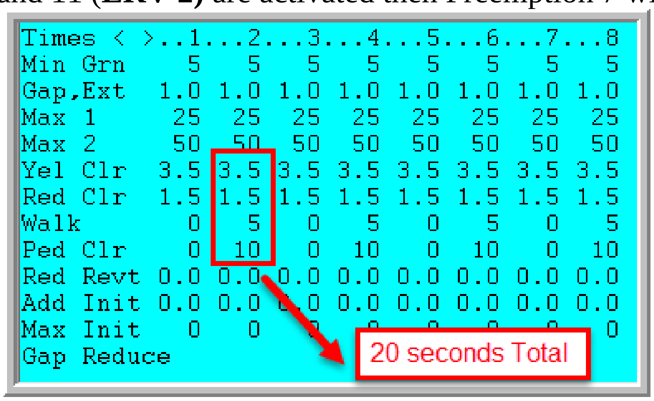
```
TimeOutPreempt  1
HoldPhases     1         2         3
     12345678 90123456 78901234 56789012
     .X...X.. ........ ........ ........
```

For each Transit Light Rail Detector (**TLRD**) (**MM->5->9->8**), the software first looks at the **TLRMode** to determine if the **TLRD** Detector is running **TLR** or **PSD**. For each **TLRD**, the software will attempt to run the programmed preemption associated with it based on the selected mode. If there is no preemption programmed, then a preemption is **not** selected to run.

If multiple **TLRD**'s are actuated at the same time, the software will instead match the **TLRD** detection selection to the matrix (**MM->5->9->5**) and chooses the programmed preemption based on exactly matching the **TLRD** Detection Selection along with preemption priority hierarchy. If there is no match, then a preemption is **not** selected.

Below are different scenarios based on the programming screens above.

1) Scenario 1: **TLRD** #1 using **TLR**. Check-in Detector 9 is activated to ON. It will result Preemption 9 being run. Detector 9 is then actuated to an OFF state. Preempt 9 will run until Check-out Detector 10 is actuated ON then OFF or the **MaxDUR** timer of 30 seconds is reached.
2) Scenario 2: **TLRD #2** using **TLR**. Check-in Detector 11 is activated to ON. It will result Preemption 10 being run. Detector 11 is then actuated to an OFF state. Preempt 10 will run until Check-out Detector 12 is actuated ON then OFF or the **MaxDUR** timer of 30 seconds is reached. In addition, **TLRD** #2 cannot run for 3 minutes due to the **L/O Time** setting.
3) Scenario 3: Both **TLRD #1** and **TLRD #2** are called at the same time. Check-in detectors 9 (**LRV 1**) and 11 (**LRV 2)** are activated then Preemption 7 will be run.



| Times < > | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Min Grn | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| Gap,Ext | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| Max 1 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 |
| Max 2 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 |
| Yel Clr | 3.5 | 3.5 | 3.5 | 3.5 | 3.5 | 3.5 | 3.5 | 3.5 |
| Red Clr | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| Walk | 0 | 5 | 0 | 5 | 0 | 5 | 0 | 5 |
| Ped Clr | 0 | 10 | 0 | 10 | 0 | 10 | 0 | 10 |
| Red Revt | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Add Init | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Max Init | 0 | 0 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 0 |
| Gap Reduce | | | | | | | | |



4) Scenario 4: **TLRD #3** using **PSD**. The Preemption Service Delay (**PSD**) Mode is intended for use with an input coming from external source like a rail cabinet or GPS. It is automatically calculated by the software. Please note that the **PSD** time is calculated to be equal to the longest programmed **walk + pedestrian clearance + yellow clearance + red clearance times.** In this case, as derived from the phase timing screen above, it will be calculated as 20 seconds. Advanced Detector 13 is activated to ON. This will put intersection in **Free and** places inhibits on all pedestrian phases immediately. It will then use the calculated **PSD** time to delay the operation of the preemption, then call Preemption 3 until the Check-Out Detector 14 is actuated ON then OFF or the 30 second **MaxDur** time expires.

5) Scenario 5: **TLRD #4** using **PSD**. Using the Timing screen above, the **PSD** time will be calculated as 20 seconds. Advanced Detector 15 is activated to ON. This will put intersection in **Free and** places inhibits on all pedestrian phases immediately. It will then use the calculated **PSD** time to delay the operation of the preemption, It is expecting a call from Check-in Detector # 16 to call Preemption 4. If Check-in detector #16 does not receive a call, then Preemption 4 will still be run after the calculated **PSD** time. The software will remain in Preemption 4 until the Check-Out Detector 17 is actuated ON then OFF or the 30 second **MaxDur** time expires.
6) Scenario 6: Both **TLRD #3** and **TLRD #4** are called at the same time. Using the Timing screen above, the **PSD** time will be calculated as 20 seconds. Advanced Detectors 13 and 15 are activated to ON. This will put intersection in **Free and** places inhibits on all pedestrian phases immediately. It will then use the calculated **PSD** time to delay the operation of the preemption, Preemption 8 will still be run after the calculated **PSD** time. The software will remain in Preemption 8 until the Check-Out Detector 14 and 17 are actuated ON then OFF or the 30 second **MaxDur** time expires.
7) Scenario 7: **TLRD #5** using **TLR.** Check-in Detector 18 is activated to ON. It will result Preemption 5 being called and run. If the controller is running phases 2 & 6, those phases will be held ON until the preemption is run. Detector 18 is then actuated to an OFF state. Preempt 5 will run until Check-out Detector 19 is actuated ON then OFF or the **MaxDUR** timer of 30 seconds is reached.
8) Scenario 8: Scenario 1: **TLRD** #1 using **TLR**. Check-in Detector 9 is activated to ON. It will result Preemption 9 being run. Detector 9 is then actuated to an OFF state. Preempt 9 will run until Check-out Detector 10 is actuated ON then OFF or the **MaxDUR** timer of 30 seconds is reached. If the **MaxDUR** timer of 30 seconds is reached and the Check-in detector 9 call remains constantly on, then Preemption 1 will be run based on the **TimeOutPreempt** programming under **MM->5->9->5**.

## Transit Preemption Matrix example

### For V80.5E or earlier and Scout V85.1 or earlier

The following example will discuss the programming of both the TranDet (MM->5->9->8) and TranPreMtrxDet (MM->5->9->5) screens.

```
     Transit Light Rail Detectors
LRV Det  ..1...2...3...4...5...6...7...8
AdvDet      0   0   0   0   0   0   0   0
ChkInDet   31  32  33  34  35   0   0   0
ChkOutDet   0   0   0   0   0   0   0   0
MaxChkIn    0   0   0   0   0   0   0   0
ChkInDly    0   0   0   0   0   0   0   0
TspMode     .   .   .   .   .   .   .   .
L/OTime     0   0   0   0   0   0   0   0
OutTime     0   0   0   0   0   0   0   0
Overlap     0   0   0   0   0   0   0   0
Preempt     0   0   0   0   0   0   0   0
UseHold     .   .   .   .   .   .   .   .
```

```
Pre Det.1.2.3.4.5.6.7.8
 1      X . . . . . . .
 2      . X . . . . . .
 3      . . X . . . . .
 4      X X . . . . . .
 5      . X X . . . . .
 6      X . X . . . . .
 7      X X X . . . . .
 8      . . . X . . . .
 9      . . . . X . . .
10      . . . X X . . .
 0      . . . . . . . .
 0   +  . . . . . . . .
```

As an example, the user has programed the standard preemption hierarchy where HP1 overrides HP2, overrides HP3, etc. Below are screens for MM->5->9->8, the Transit Check-in detectors, and MM->5->9->5 the Transit Preemption detector Matrix:

The software matches the LRV detection selection to the matrix and chooses the programmed preemption based on exactly matching the LRV Detection Selection along with preemption priority hierarchy. If there is no match then a preemption is not selected.

1) Scenario 1: Detector 35 (LRV 5) is activated will result in Preemption 9 being run.
2) Scenario 2: Both Detector 31 (LRV 1) and 32 (LRV 2) are activated then Preemption 4 will be run.
a. While Preemption 4 is running, the LRV 1 call drops so there is only a LRV 2 call, then Preemption 2 will run because it is a higher priority.
3) Scenario 3: Detector 31 (LRV1), Detector 34 (LRV 4) and Detector 35 (LRV 5) are activated, no preemption will be selected because that combination of LRV detectors are not defined in the matrix.

## Transit Preempt Service Delay (PSD) using V80.x/V85.x controller software

**NOTE: Trafficware would like to thank Fred Mills of the City of Houston and Mike Taylor of Taylor Traffic for the write-up below discussing PSD.**

To understand **PSD**, consider the example below without preempt service delay.

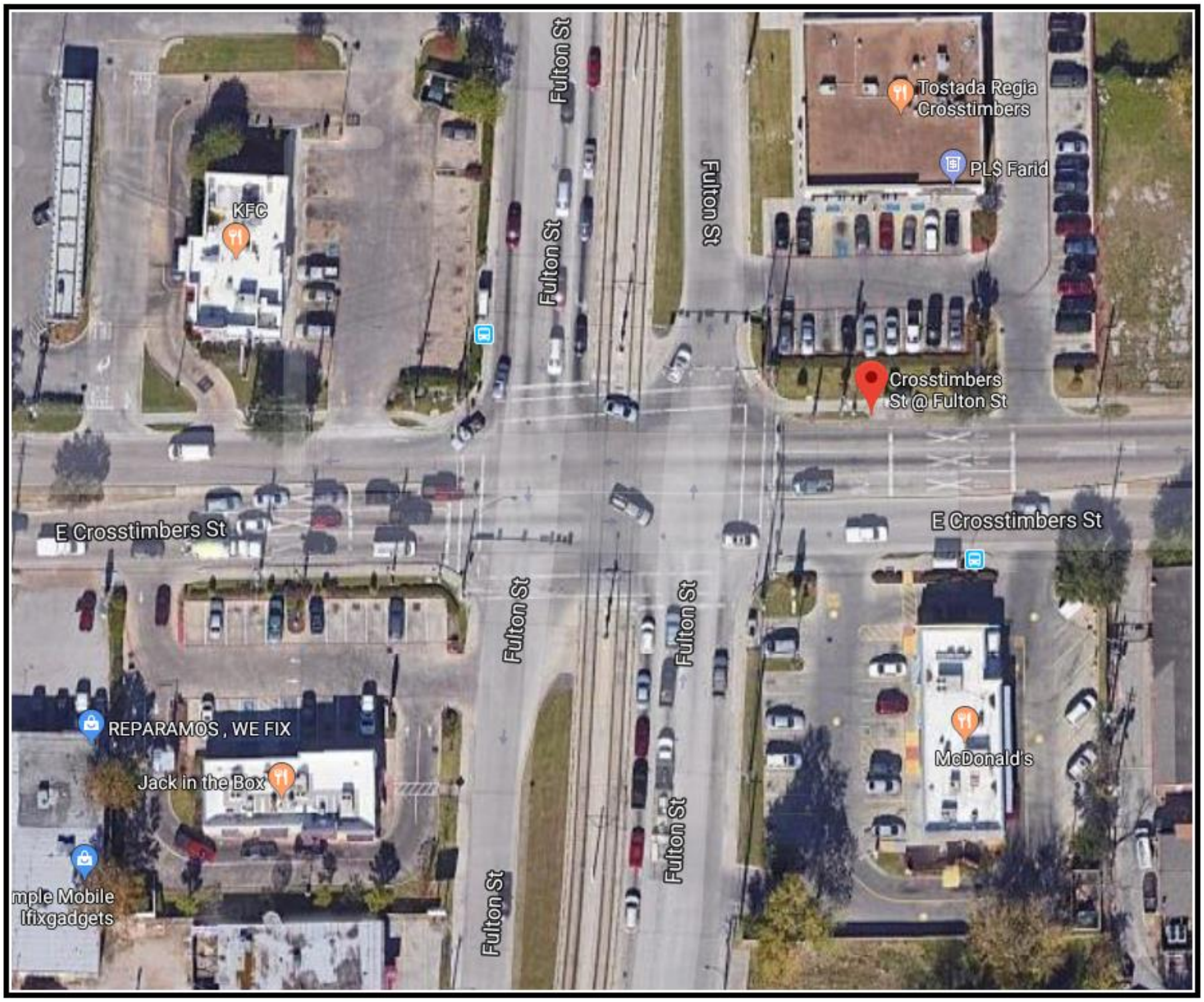
Consider two adjacent intersections. The first intersection started a conflicting pedestrian phase immediately prior to receiving an LRT preempt input. The second intersection is at a portion of the cycle where it can respond immediately to the LRT preempt input. The second intersection will dwell for **PY** plus the travel time prior to the light rail vehicle arriving at the intersection. While this functions to keep the light rail vehicle moving, it is very inefficient for other road users and frequently results in citizen complaints. In heavy traffic conditions the extra delay at the second intersection can cause spill back at other adjacent intersections. **PSD** provides the same level of service to the light rail vehicle without the inefficiency of traditional preemption inputs. The intersection keeps cycling during **PY** time prior to starting the preempt, and each phase or pedestrian signal that could delay the start of the preempt is inhibited at the point it would impact the start of the preempt.

Preemption service must consider the situation where the light rail train does not arrive in a reasonable amount of time. If the vehicle for which priority is being provided fails to use the priority in a reasonable amount of time, the intersection should resume cycling for other users. The value of "reasonable amount of time" is a discretionary consideration and will need to be set for each system. If the input does not drop (indicating the train has cleared the intersection) within the "reasonable amount of time" it is considered to have timed-out/failed. Each preempt input has a user definable time-out value. Once the LRT input has timed-out, and no other non-timed-out LRT inputs are on, the intersection will run the time-out preempt. Note that there can be instances where the first LRT input is timed out, but a second LRT input started prior to the first timing-out. In this instance the controller will time the preempt for the second input prior to entering the time-out preempt.

The time-out preempt returns the intersection to cycling. All phases and overlaps are enabled in the time-out preempt. The intersection runs free and uses the values from MM->1->1->1. The time-out preempt remains running until all LRT inputs drop. The City of Houston normally sets relatively short min and max green times for time-out, so the intersection will cycle quickly. Typical starting values are 10 seconds max green for LRT phases, 15 seconds max green for left turn and side street phases, and 25 seconds max green for major street phases. One key practice is to set the min green for LRT phases to zero, this prevents conflicting outputs when a train clears the intersection during time-out and the controller exits preemption. During fail-safe preempt operation no coordination is running.

Two additional features are required to get the best use from **PSD**: priority hold phases and dynamic exit. Dynamic exit instructs the traffic controller at the end of a preempt to service the phase that has been waiting the longest. Dynamic exit is available to all preempts in Trafficware V80.x and V85.x/Scout software and is not specific to **PSD** operations. Priority hold phases instruct the controller to remain in a designated phase until the start of the **PSD** preempt if the controller is currently in that phase. Priority hold phases are used at locations where the majority of traffic is on approaches that conflict with the light rail approaches. Defining priority hold phases allows a corridor to remain green until forced off by the **PSD** preempt.

## Transit Preempt Service Delay (PSD) Example

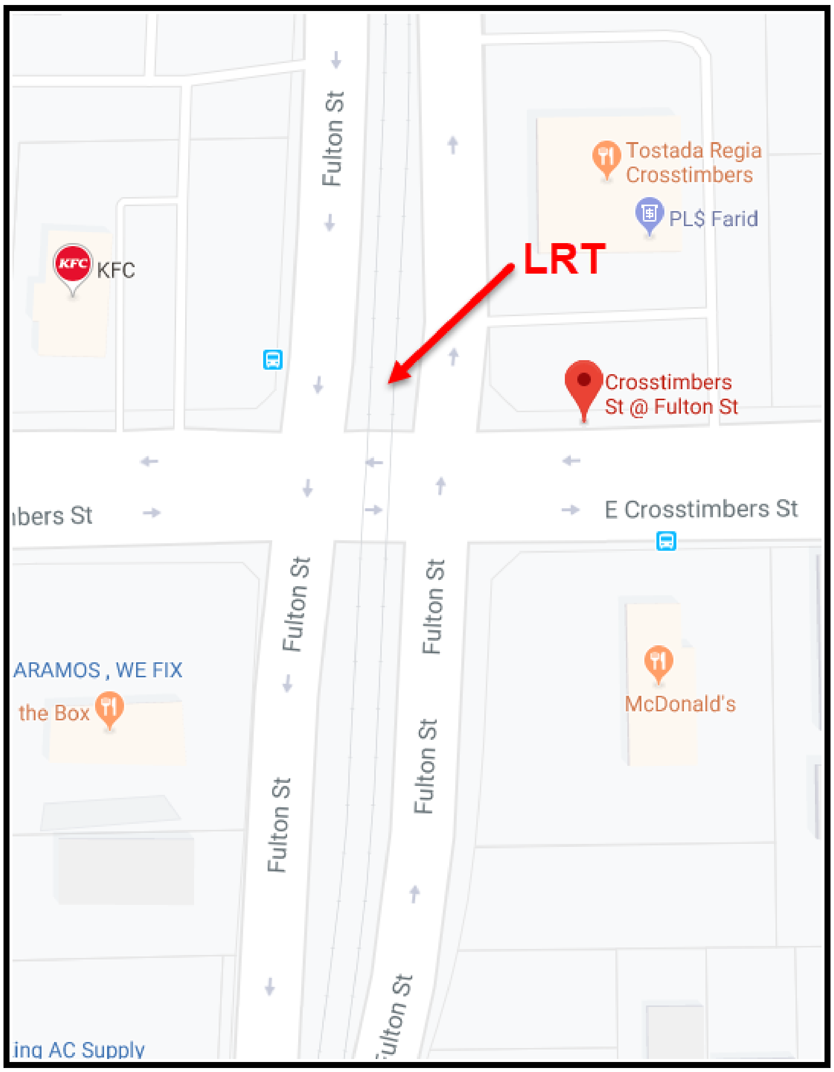



The example below utilizes priority hold phases.

In order to provide a better understanding of **PSD** an example is provided here. Your authors' opinions are that a well described real-world example provides more clarity than pages of mathematical formulas. Here we will describe **PSD** at an eight-phase intersection, specifically the Houston intersection of Crosstimbers @ Fulton, as shown as above. This example is not intended to be an exhaustive detailing of every aspect of this intersection, instead it is intended to provide an understanding of **PSD** at a typical LRT signalized intersection. **PSD** can be utilized at much more complex intersections (such as the **PSD** implemented at Emancipation @ Harrisburg @ Texas), but the complicated ring structure of a more unusual intersection is not necessary for a good example.

Therefore, we have chosen an intersection type that, other than the LRT components, should have a familiar configuration to everyone in the traffic signal industry.

Crosstimbers is an East / West road that is is the major corridor. Fulton is a North / South road and is the side street. The left turns are signalized. Fulton has two sets of light rail tracks running in the median north / south through the intersection. At this location light rail trains can travel either direction, north or south, on either track. In addition to the normal vehicle and pedestrian signals there are two light rail “bar” signals, one each for northbound LRT and for southbound LRT. The northbound bar signal applies to northbound trains on either track and the southbound signal applies to southbound trains on either track. The only other noteworthy configuration feature of this intersection is that the north and south left turn paths conflict due to the wide median. Therefore, the phasing sequence and concurrencies need to prevent these lefts from running simultaneously.



This intersection can be run coordinated in STD8 mode with only phases 1 through 8 enabled. The ring structure and concurrencies require modifications. The normal 8 phase ring sequence is:

| | | | | |
|---|---|---|---|---|
| Ring 1 | Φ 1 | Φ 2 | Φ 3 | Φ 4 |
| Ring 2 | Φ 5 | Φ 6 | Φ 7 | Φ 8 |
| Barrier Group | A | | B | |

For Crosstimbers at Fulton the "B" barrier group sequences need to be changed to always lead-lag or lag-lead the north and south lefts. Also, phasing needs to be included for the light rail. The light rail approach adjacent to the vehicle through is phased as the normal direction +10; that is the light rail signal adjacent to phase 4 is driven by phase 14 and adjacent to phase 8 is driven by phase 18. Thus, our ring structure and concurrencies become:

| | | | | | |
|---|---|---|---|---|---|
| Ring 1 | Φ 1 | Φ 2 | Φ 3 | Φ 14 | Φ 4 |
| Ring 2 | Φ 5 | Φ 6 | Φ 8 | Φ 18 | Φ 7 |
| Barrier Group | A | | | B | |

OR

| | | | | | |
|---|---|---|---|---|---|
| Ring 1 | Φ 1 | Φ 2 | Φ 4 | Φ 14 | Φ 3 |
| Ring 2 | Φ 5 | Φ 6 | Φ 7 | Φ 18 | Φ 8 |
| Barrier Group | A | | | B | |

The "A" barrier group concurrencies remain unchanged and the sequences can have any combination of lead-lead, lead-lag, lag-lead, or lag-lag. The "B" barrier group concurrencies are:

| Phase | Concurrent Phase (s) | | |
|---|---|---|---|
| 3 | 8 | | |
| 4 | 7 | 8 | 18 |
| 7 | 4 | | |
| 8 | 3 | 4 | 14 |
| 14 | 8 | 18 | |
| 18 | 4 | 14 | |

This ring structure allows for normal 8 phase operation, for individual northbound or southbound trains, and for northbound and southbound simultaneous trains.

Overlaps are needed for this location. The parallel northbound and southbound vehicle signals can remain green while a train passes through the intersection. Overlaps 4 and 8 are defined as normal overlaps with parent phases 4 & 14 and 8 & 18 respectively. The bar signals use **GoBar** type overlaps 14 and 18 with one parent phase each, 14 and 18 respectively.

As mentioned previously phases 1 through 8 are enabled in the MM->1->1->2 screen while phases 14 & 18 are not enabled. The light rail phases 14 or 18 only are enabled when called by a preempt. This allows for the coordination and other operational checks for a standard 8 phase intersection during coordination and provides light rail operation on demand.

Preempts 3 through 6 are set up as to provide traditional emergency vehicle service and are not further discussed here. Preempt 8 is set up to provide simultaneous northbound and southbound light rail and has dwell phases 14 and 18. Preempt 11 is set up to provide northbound light rain and has dwell phases 4 and 18. Preempt 12 is set up to provide southbound light rail and has dwell phases 8 and 14.

Staggered LRT inputs will result in an individual LRT preempt running, then the preempt for both directions, then the other individual LRT preempt. For example, a northbound input followed by a southbound input will result in preempt 11, then preempt 8, then preempt 12 (assuming the northbound input drops before the southbound input).

Crosstimbers through traffic is the majority of the demand at this intersection. Eastbound and westbound phases 2 and 6 are flagged as priority hold phases. If the controller starts phase 2 and/or phase 6 during **PY** time then that phase (or both phases) will remain green regardless of max times, split times, etc. until forced off by the start of the **PSD** preempt.

## Programming PSD

Few screens and settings are required to program **PSD**. **PSD** as implemented by Trafficware works with a latched input. To clarify, the input is turned on and stays on until LRT service is no longer needed. Logic could be used to create a latched input where pulsed inputs are provided to the traffic cabinet. The steps to program **PSD** in the example above are:

Assign the appropriate LRT inputs to detectors in MM->5->1. Set the fail time to 255 on each LRT detector. In our example we set detector 65 to phase 18, 66 to phase 14, 69 to phase 18, and 70 to phase 14.

In MM->5->2 set the LRT detectors to call and extend. In our example we set detectors 65, 66, 69, and 70 to call and extend only.

In MM->5->9->8 set the LRV detectors to call the appropriate preempt. The following settings are used for LRT detector 1 (left column):

Transit Light Rail Detectors

| LRV Det | ..1 | ...2 | ...3 | ...4 | ...5 | ...6 | ...7 | ...8 |
|---|---|---|---|---|---|---|---|---|
| AdvDet | 65 | 66 | 68 | 70 | 0 | 0 | 0 | 0 |
| ChkInDet | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ChkOutDet | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MaxDur | 30 | 30 | 30 | 0 | 0 | 0 | 0 | 0 |
| ChkInDly | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TLRMode | . | . | . | . | X | X | X | X |
| L/OTime | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| OvrTime | 84 | 84 | 84 | 84 | 0 | 0 | 0 | 0 |
| Reserved | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Preempt | 11 | 12 | 11 | 12 | 0 | 0 | 0 | 0 |
| UseHold | X | X | X | X | . | . | . | . |

All settings not explicitly discussed in this list are left unassigned or set to 0.

AdvDet set to 65. This ties LRT detector 1 to vehicle detector 65.

MaxDur is set to 30. This is the number of seconds the preempt dwell phase will run before the input times-out.

OvrTime is set to 84. This provides a maximum of 84 seconds of preemption if the override input for LRT 1 is on. 84 is MaxDur time plus **PY** time (30+53.9 for this example).

Preempt is set to 11.

UseHold is flagged (X). This turns on the use of priority hold phases for this LRT input.

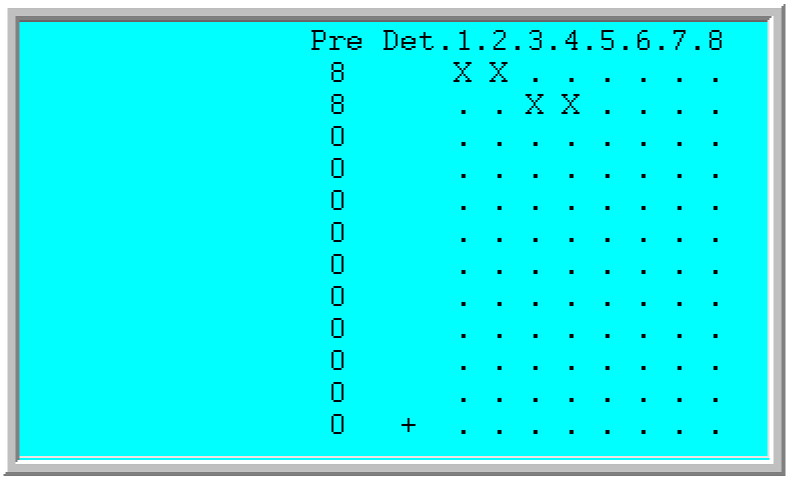
In MM->5->9->8 LRV detector 2 is set the same as above, with two exceptions:

AdvDet is set to 66

Preempt is set to 12

LRV detectors 3 and 4 are set for AdvDet 68 and 70, and call Preempts 11 and 12 respectively. Otherwise LRV detectors 3 and 4 are set the same as LRT detector 1.

In MM->5->9->5 we allow for combinations of light rail inputs. The left column labeled **Pre** designates which preempt is called. The detection matrix is used to assign which combinations of LRV detector will call the preempt:




In our example we set row 1 Pre to 8 and flag LRV detectors 1 and 2. This calls the combined north and south LRT preempt when both LRV detectors 1 and 2 are on at the same time.

Likewise we set row 2 to call Preempt 8 and flag LRV detectors 3 and 4. This calls the combined north and south LRT preempt when both LRV detectors 3 and 4 are on at the same time.

## Transit Preempt Service Delay (PSD) Mathematical Description

Preempt service delay is accomplished by incrementally limiting the vehicle and pedestrian movements that can run between the time an LRT request is received and when the associated train signal upgrades.

Each vehicle phase and pedestrian signal that conflicts with a bar signal has an individual yield time. Once each of these signals starts there is a minimum time that must pass before the bar signal can run. Each of these yield times can be calculated as follows:

Yield(Ø x) = minGreen(Ø x) + amber clearance(Ø x) + red clearance(Ø x)

Yield(ped x) = walk(Ø x)+ flashing don^' t walk(Ø x) + amber clearance(Ø x) + red clearance(Ø x)

x=1,2,3…for all conflicting phases and peds

**PY** = Max Yield Time

In practice **PY** time for all the intersections is the yield time of the longest pedestrian movement crossing the tracks. This has been consistent enough that this longest yield time came to be called Pedestrian Yield time, then was shortened to **PY** time. The signal timing engineer calculates **PY** time and provides that number to the light rail operator. The light rail operator sends the traffic case an input **PY** time before the LRT signal needs to upgrade.

All modern signal controller software options that your author is aware of have similar basic preemption operation. For phases not included in each preempt there are programmable values for minimum green, walk, and ped clearance. For this discussion the phases not included in a preempt are conflicting phases. The traffic controller will run the following sequence when a preempt input is activated:

1) Honor the programmed min walk for any conflicting peds running. When there are no conflicting peds running this time is zero.

2) Honor the programmed ped clearance for any conflicting peds running. When there are no conflicting peds running this time is zero.

3) Honor the programmed min green for any conflicting phases. When there are no conflicting phases running this time is zero. Note that if a ped is running the ped times are almost guaranteed to exceed the phase min green.

4) Force Off the conflicting phase(s) green(s)

5) Run the conflicting vehicle clearances (yellow and all red).

6) Run the phases called by the preempt

Note that the final portion of **PY** is the vehicle clearances caused by the preempt. Therefore, Preempt Apply Time can be defined as **PY** minus the largest conflicting vehicle clearances

PAT = **PY**-max$f()$(clearances)

When an LRT request is received each phase or ped needs to be prevented from starting after their yield time. This can be accomplished by applying an inhibit to the ped or phase. If a phase is running, applying an inhibit to it does not change operations. If a phase is not running the clearances for the phase running must be timed before the next phase can start. This affects when to omit phases. The omit must be applied before the preceding phase clearances start. If a timer is started when the request is received these times will be:

Inhibit (ped x or Ø x) = PAT (intersection)- Yield (ped x or Ø x)

Note that the largest ped yield time equals the intersection **PY** time, thus that delay value is negative and that ped needs to be omitted immediately. Phases or peds with large min green or ped times can have negative omit times, they should also be omitted immediately. As the timer counts, more phases and peds will be inhibited. By the point that the timer is equal to PAT the only phases and peds not inhibited are those that can run concurrently with LRT operation. Note that an inhibit prevents the start of a phase or ped only. Inhibit does not force a phase or ped to leave green or walk.

For Preempt Service Delay to work the intersection must be in a state where minimum green, walk, and ped clearances have been satisfied at PAT after a request is received. If the ped crossing the tracks is in walk when the request is received, then that walk needs to be terminated (honoring the min walk) and the ped clearances run or **PY** time will be exceeded. For other peds and vehicle phases this can be accomplished by applying of inhibits described in the previous paragraphs.

Be aware that in the phase yield time equations the min green for a phase needs to be set carefully. Too large a value of min green has the potential to prevent the phase from running before the LRT preempt. Too small a value of min green doesn't move many vehicles and frustrates drivers. This author starts with the following min green values and adjusts based on field observation: left turns with moderate volume min green 7 seconds; left turns with heavy volume min green 12 seconds; cross street with moderate volume min green 8 seconds; cross street with heavy volume min green 15 seconds.

## Summary

The Preempt Service Delay (**PSD**) works by eliminating most of the inefficiencies at light rail intersections that occur when using preemption.