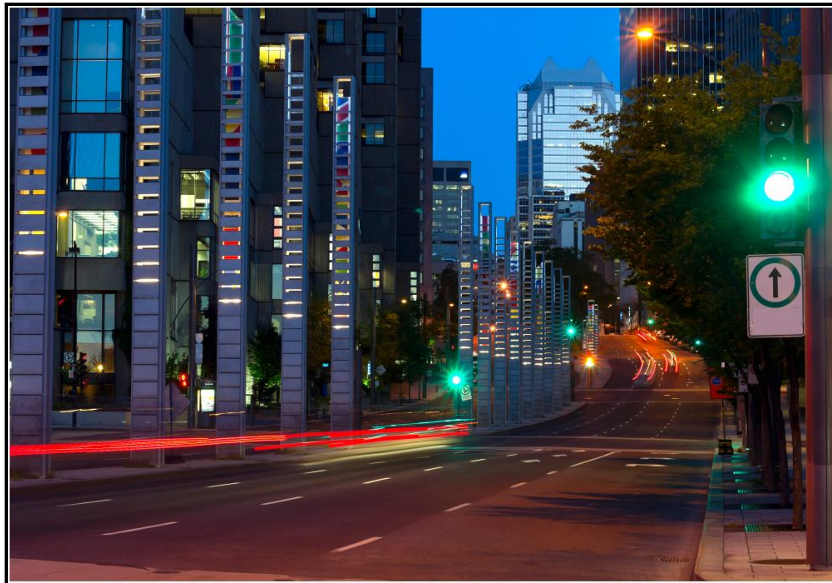




## *Operations Manual*

*For*

# **Advanced Coordination Topics for Controllers Using V85.x/Scout**



April 2021

522 Gillingham  
Sugar Land, Texas 77478  
Phone: (281) 240-7233  
Fax: (281) 240-7238

© Copyright 2021 Cubic | Trafficware  
All rights reserved.

# Terms of Use Agreement

## USE OF THIS SOFTWARE MANUAL IS SUBJECT TO THESE TERMS OF USE.

### Agreement

Please read this *User Agreement* carefully before using the information provided in this *Manual*. This Agreement explains the terms and conditions governing the use of this *Manual* ("Terms of Use"), and it is your responsibility to read and understand them. By using this *Manual*, you expressly agree to be bound by these *Terms of Use* and to follow them as well as all applicable laws and regulations governing the *Manual*. If you do not agree to be bound by these *Terms of Use*, you may not access or use this *Manual*. Cubic | Trafficware reserves the right to change these *Terms of Use* at any time, effective immediately upon posting the *Manual* from our company website. By continuing to use the *Manual* after we post any such change, you accept the revised *Terms of Use*. If you violate these *Terms of Use*, Cubic | Trafficware may terminate your use of the *Manual*, bar you from future use of the *Manual*, and take appropriate legal action against you.

### Permitted Use

You agree that you are only authorized to read, view and retain a copy of pages of this *Manual* for your own personal use, and that you will not duplicate, download, publish, modify or otherwise distribute the material on this *Manual* for any purpose other than to review product information for personal use or the use of a government or non-profit organization.

### No Commercial Use

Users may not use this *Manual* for any commercial purposes such as to sell merchandise or services of any kind. You must obtain our prior written consent to make commercial offers of any kind, whether by advertisements, solicitations, links, or by any other form of communication. Cubic | Trafficware will investigate and take appropriate legal action against anyone who violates this provision.

### Copyright

All content included on this *Manual*, including text, graphics, logos, icons, images, and software is the property of Cubic ITS Inc. or its content suppliers and is protected by United States and international copyright laws. This compilation (that is, the collection, arrangement and assembly) of all content on this *Manual* is the exclusive property of Cubic ITS Inc. and is protected by U.S. and international copyright laws. Cubic ITS Inc. reserves the right to revise the pages of the *Manual* or withdraw access to them at any time.

### Trademarks

The logo and trademarks that appear throughout the *Manual* belong to Cubic ITS Inc., its affiliates or third-party trademark owners, and are protected by U.S. and international trademark laws. Without express prior written permission, you may not display or use in any manner, the logos or trademarks. Nothing in this *Manual* shall be construed as granting any license or other right to the intellectual property or other proprietary rights of Cubic ITS Inc., its affiliates or any third party, whether by estoppels, implication or otherwise. All contents of the *Manual* are: © Copyright 2021 Cubic ITS Inc. or its licensors. All Rights Reserved.

### Disclaimer of Warranty

YOU UNDERSTAND AND EXPRESSLY AGREE THAT YOUR USE OF THE MANUAL AND THE INFORMATION FOUND THERE IS ENTIRELY AT YOUR RISK. CUBIC | TRAFFICWARE AND ITS AFFILIATES AND LICENSORS MAKE NO WARRANTIES OR ANY REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF TITLE OR NON-INFRINGEMENT OR IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OR OTHER VIOLATION OF RIGHTS IN RELATION TO THE AVAILABILITY, ACCURACY, VALIDITY, COMPLETENESS, RELIABILITY OR CONTENT OF THESE PAGES AND/OR THE MANUAL. CUBIC | TRAFFICWARE SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR FOR BUSINESS INTERRUPTION ARISING OUT OF THE USE OF OR INABILITY TO USE THIS MANUAL, EVEN IF CUBIC | TRAFFICWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. TO THE MAXIMUM EXTENT PERMITTED BY LAW, YOU HEREBY RELEASE AND FOREVER WAIVE ANY AND ALL CLAIMS YOU MAY HAVE AGAINST CUBIC | TRAFFICWARE, ITS AFFILIATES AND LICENSORS FROM LOSSES OR DAMAGES YOU SUSTAIN IN CONNECTION WITH YOUR USE OF THE MANUAL.

SOME JURISDICTIONS DO NOT ALLOW EXCLUSION OF CERTAIN WARRANTIES OR LIMITATIONS OF LIABILITY, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU. THE LIABILITY OF CUBIC | TRAFFICWARE WOULD IN SUCH CASE BE LIMITED TO THE GREATEST EXTENT PERMITTED BY LAW.

### Applicable Law

These *Terms of Use* and all legal issues related to the *Manual* shall be governed by the laws of the State of Texas, without regard to conflict of laws principles. You agree that any dispute involving these terms of use or this *Manual* will be heard in a court with jurisdiction in Fort Bend County, Texas. You further agree that the prevailing party in any legal action will be entitled to recover its reasonable attorney's fees incurred in connection with that action. If for any reason a court of competent jurisdiction finds any provision of these Terms of Use to be unenforceable, the remaining terms will continue in full force and effect.

### Entire Agreement

These *Terms of Use* are the final and entire agreement between you and Cubic | Trafficware with respect to this subject and replace any and prior or contemporaneous understandings or agreements, written or oral, regarding the subject matter. Any waiver of any provision of these Terms of Use shall be effective only if in writing and signed by an authorized representative of Cubic | Trafficware.

# Table of Contents

<b>1</b>	<b>COORDINATION PROGRAMMING CONSIDERATIONS AND BEST PRACTICES .....</b>	<b>5</b>
1.1	GENERAL CONCEPTS USING NTCIP COORDINATION.....	6
1.1.1	Permissive window (NTCIP FIXED and FLOAT):.....	6
1.1.2	Stop in Walk.....	6
1.1.3	Min Perm V/P: .....	7
1.1.4	PedCall.....	7
1.1.5	Transition.....	7
1.1.6	Return Hold.....	8
1.2	STANDARD COORDINATION CALCULATIONS .....	9
1.2.1	Minimum Time Requirement .....	9
1.2.2	Permissive window .....	9
1.2.3	Transition calculations .....	12
1.3	COORDINATION MODIFICATIONS IN CONTROLLERS WITH V85.2 OR LATER .....	14
1.3.1	Permissive window Modifications .....	14
1.3.2	Transition Modifications.....	16
1.4	COORDINATION MODES AND MODES PLUS (MM->2->1) .....	19
1.4.1	Correction Mode .....	19
1.4.2	Force-Off Mode.....	19
1.4.3	Maximum Mode (only used during FIXED).....	19
1.4.4	Stop-in Walk.....	20
1.4.5	Walk Recycle.....	20
1.4.6	Leave Walk Before / Leave Walk After .....	22
1.4.7	NTCIP Yield.....	22
1.4.8	Early Yield (EarlyYld) .....	23
1.4.9	FreeOnSeqChg.....	23
1.4.10	No Added Initial .....	23
1.4.11	PedCallInh.....	23
1.4.12	MinPermV/P.....	23
1.4.13	Coord Hold .....	24
1.5	EASY CALCS STATUS SCREEN (MM->2->8->2) FOR CONTROLLERS USING SCOUT/V85.2x OR LATER	25
1.5.1	Primary Force-Off.....	25
1.5.2	Secondary Force-Off.....	25
1.5.3	Vehicle Yield .....	25
1.5.4	Vehicle Apply .....	25
1.5.5	Pedestrian Yield .....	25
1.5.6	Ped Apply.....	26
1.5.7	FloatMx .....	26
1.5.8	PedLeav .....	26
1.5.9	PedCall.....	26
1.5.10	SplitRem.....	26
1.6	EASY CALCS STATUS SCREEN (MM->2->8->2) FOR CONTROLLERS USING SCOUT/V85.1x .....	27
1.6.1	Primary Force-Off.....	27
1.6.2	Secondary Force-Off.....	27
1.6.3	Vehicle Yield .....	27
1.6.4	Vehicle Apply .....	27
1.6.5	Pedestrian Yield .....	27
1.6.6	Ped Apply.....	28
1.6.7	FloatMx .....	28
1.6.8	PedLeav .....	28

1.6.9 PedCall.....	28
1.6.10 SplitRem.....	28
1.7 TRANSITION CONSIDERATIONS USING NTCIP SETTINGS .....	29
1.7.1 Defining transition .....	29
1.7.2 Real Time Status screens to Monitor .....	30
1.7.3 Specific settings that affect transitions.....	31
1.8 PERM,FRC (MM->2->7->2).....	33
1.8.1 Primary Force-Off.....	33
1.8.2 HoldToMax .....	33
1.8.3 VApply .....	33
1.8.4 VehYld .....	34
1.8.5 PApply .....	34
1.8.6 Pedestrian Yield .....	34
1.8.7 Permissives .....	34
1.8.8 FrcAll.....	34
1.8.9 PedRcy.....	34
1.8.10 OlpPedRcl.....	35
1.8.11 Mx/Frc .....	35
1.9 COORDINATION ALARM CONSIDERATIONS.....	36
1.9.1 Algorithmic details of various coordination alarms .....	37
1.9.2 Alarm 17: Cycle Fault .....	40
1.9.3 Alarm 30 Pattern Error Faults.....	40
1.10 USER MODE DIAGNOSTICS .....	41
1.11 STANDARD 8 PHASE OPERATION VERSES USER MODE OPERATION- GENERAL CONSIDERATIONS.....	42
1.12 COORDINATION EXAMPLES .....	43
1.12.1 Basic program.....	43
1.12.2 Ped Recycle Considerations.....	45
1.12.3 Stop-in walk considerations. ....	46
<b>2 OTHER COORDINATION MODES.....</b>	<b>49</b>
2.1 EASY.....	49
2.2 PERM,FRC .....	50
2.2.1 Primary Force-Off.....	50
2.2.2 HoldToMax (V80.x or V85.x/Scout only).....	50
2.2.3 VApply .....	51
2.2.4 VehYld .....	51
2.2.5 PApply .....	51
2.2.6 Pedestrian Yield .....	51
2.2.7 Permissives .....	52
2.2.8 FrcAll.....	52
2.2.9 PedRcy.....	52
<b>3 EXTERNAL INPUT / OUTPUT PROGRAMMING.....</b>	<b>53</b>
3.1 SETTING UP EXTERNAL I/O .....	53
3.2 EXTERNAL PLAN SETUP .....	54
3.3 EXTERNAL I/O PROGRAMMING STEPS.....	54
3.3.1 General Information.....	54
3.3.2 Coordination Menu (MM->2).....	54
3.3.3 Plan Selection Examples.....	55

# 1 Coordination Programming Considerations and Best Practices

NTCIP Coordination at its best allows users to simply program coordination for a corridor of intersections. As explained in Chapter 6 of the V85.x/Scout NTCIP controller manual description, the user simply programs patterns, cycle lengths, offsets, phase splits and various coordination mode general features. The NTCIP coordination algorithms developed by Cubic | Trafficware will then automatically calculate all necessary constructs to run coordination. This section will describe considerations and best practices that users should review to efficiently run coordination on a set of intersections.

**Note:** When considering coordination, using the STD8 phase mode will take advantage of the most coordination diagnostic checks to catch common data entry mistakes, and if detected, times the intersection in FREE. In USER mode, most of these coordination diagnostics are removed, and the onus is on the agency verify and test the programming to ensure that coordination pattern(s) run as expected.

[V85.2] NOTE: Patterns numbers and Split Table numbers have been increased from 48 to 253.

Pattern#	Coord	FreeStat	Active State of the Coordinator
0	FREE	PATTERN	Coordinator has selected default free pattern# 0 by time-of-day
1 - 253	ACTIVE	CoorActv	Coordinator is running one of the 253 patterns under coordination
1 - 253	FREE	COMMAND	Coordinator is running one of the 253 patterns in free operation
254	FREE	COMMAND	Coordinator is running the NTCIP Free Pattern# 254
255	FREE	COMMAND	Coordinator is running the NTCIP Flash Pattern# 255

**NOTE:** The features explained below may not be available for all versions of software and may be subject to adding additional Modules or Licensing to your controller software. Please contact your Trafficware representative for further details.

# 1.1 General Concepts using NTCIP Coordination

## 1.1.1 Permissive window (NTCIP FIXED and FLOAT):

The vehicle or pedestrian permissive window is referred to the portion of the cycle during which vehicle or pedestrian calls can be served if these are calls present. The permissive window begins at the vehicle/pedestrian Yield points (***VehYield or PedYield***) and ends at ***VehApply or PedApply*** point when the phase inhibit is applied for the vehicle or pedestrian calls until the next cycle.

- **VehYield and PedYield**
  - The default Yield points for all non-coordinated phases are set at the force-off of the coord phase.
  - The default Yield point of the coord phase and the pseudo-coord phase is set 10 sec later, which delays the permissive period for the coord phase to allow “late” side street to serve after the barrier.
- **VehApply and PedApply**
  - The VehApply point is calculated by subtracting the min green time from the force-off point of the phase. This calculation is to maximize the permissive window for late vehicle calls.
  - In general cases, the PedApply point is calculated by subtracting the Walk + Ped Clearance from the force-off point of the phase. This calculation insures that the ped signal time does not cross the force-off.

## 1.1.2 Stop in Walk

Stop-in-Walk allows the programming of a phase split time to be less than the minimum pedestrian (Walk + Ped Clearance) time. Otherwise, the Coordination Diagnostics will fail and the controller will be placed into free operation. Thus, when the ped call serves with Stop-in-Walk, the walk or ped clearance may overrun the force-off, and Stop-in-Walk makes the local counter to “stop” at the force-off point if overrunning the force-off which leads to lose the coordination sync. Stop-in-Walk can be used together with the short-way offset correction to get in SYNC in one cycle.

However, Stop-in-Walk is set in **MM->2->1** and applied to all patterns. For some patterns have phases whose split times being long enough to satisfy the minimum pedestrian time, their PedApply points may be unnecessarily late.

### 1.1.3 Min Perm V/P:

These two parameters in MM->2->5 enable the minimum permissive window for vehicles and pedestrians on a pattern-by-pattern basis. This feature is usually used combined with Stop-in-Walk to prevent a late vehicle or pedestrian call from being serviced. If Min P is enabled with Stop-in-Walk, the calculation of the PedApply with Min V/P overrides that of Stop-in-Walk and thus allow the minimum permissive window for pedestrians.

With Min Perm P setting, the PedApply point is set to 5 sec after the start of the split time.

### 1.1.4 PedCall

PedCall is only used when “PedCallInh” in MM >2->1 is set to “ON” and Stop-in-Walk is not set. PedCall displays the last time a call can be placed in the cycle so a pedestrian can be serviced in that cycle.

### 1.1.5 Transition

The Programmed Offset is added to the zero point of the Time base Coordination (Tbc) counter to provide the “SYNC” point for the coord phase at Loc = 0. The controller is in SYNC when the Coord Phase (Loc = 0) is lined up with the Programmed Offset applied to the Tbc counter. Otherwise, the controller is in transition, running the local clock faster or shorter so local controller time will match the TBC time.

- **Long-way Transition**

During long way transition, the local controller clock is running slower than real time. During Long-way transition, each Split Time is lengthened by the Long-way% value programmed for the pattern. Valid values for this parameter are 0-50.

- **Short-way Transition**

During short way transition, the local controller clock is running faster than real time. The controller will shorten the split times by the Short-way% value programmed under MM->2->5 (Valid values for this parameter are 0-25). During Short-way transition, the reduced Split Times must be adequate to service the minimum phase times (*Check this requirement during Coordination Diagnostics*). Short-way is very effective with the Stop-In-Walk feature and allows the controller to transition quickly when an occasional pedestrian service extends a phase past its force-off.

- **Dwell transition**

Dwell transition is enabled for a pattern if both Short and Long values are set to zero and Dwell is set to 1-99 seconds. The Dwell method corrects the offset by resting at the end of the coordinated phase until the desired offset is reached or until the Dwell time expires. Dwell offset correction is not as popular as the short-way/long-way method.

- **Dynamic Short-way**

Dynamic Short-way transition is enabled through setting the “DynShortway” option in MM->2->1. It provides an alternative way to speed up transitions by applying an dynamically calculated rather than fixed programmed short-way percentage to shorten the split time. Dynamic Short-way correction firstly calculates the largest allowable Short-way percentage for each phase that **does not** violate the greater of the phase and pedestrian minimum time and the amount of time that the phase has used in the last cycle. Then, among the above calculated short-way percentages of currently ON phases, it chooses the smallest short-way percentage and uses that value for Shortway transitions.

### **1.1.6 Return Hold**

Return Hold programmed in MM->2->5 only applies to the coord phase is using NTCIP FIXED and FLOAT modes. When the coord phase has either a Min Recall or Max Recall set, enabling Return Hold causes the coord phase hold the green until it is forced-off. For the pseudo coord phase, if the associated through movement has a Min Recall set, then it will get the hold, otherwise the pseudo itself needs to have the min recall to get the hold. The associated through is the next high even phase number.

Disabling Return Hold allows the controller to gap-out of the coordinated phase.



## 1.2 Standard Coordination Calculations

### 1.2.1 Minimum Time Requirement

If we consider if the split time of one phase is enough for minimum pedestrian time, the minimum pedestrian time includes yellow clearance and all-red.

- **Minimum Pedestrian Time:**

If *PedClr-Thr-Yellow* is not enabled,

$\text{Ped Min} = \text{Walk} + \text{Ped Clearance} + \text{Yellow} + \text{All-Red}.$

Otherwise

$\text{Ped Min} = \text{Walk} + \text{Ped Clearance} + \text{All-Red}.$

- **Minimum Phase Time:**

$\text{Veh Min} = \text{Min Green} + \text{Yellow} + \text{All-Red}.$

### 1.2.2 Permissive window

#### **Ped Apply**

In all formulas below, Walk is the maximum time of Walk or Walk2

- **No Stop-in-Walk, no Rest-in-Walk, no Min Permissive Ped**

$\text{PedApply} = \text{PrimaryForceOff} - (\text{Walk} + \text{PedClearance})$

If *PedClr-Thru-Yellow* is enabled,

$\text{PedApply} = \text{PedApply} + \text{Yellow}.$

- **Stop-in-Walk is ON**

$\text{PedApply} = \text{PrimaryForceOff} - 5$

- **Rest-in-Walk is ON**

$\text{PedApply} = \text{PrimaryForceOff} - (\text{Walk} + \text{PedClearance})$

If *PedClr-Thru-Yellow* is enabled,

$\text{PedApply} = \text{PedApply} + \text{Yellow}.$

- **MinPerm Ped is ON**

$\text{PedApply} = \text{PrimaryForceOff} + \text{Yellow} + \text{All-Red} - \text{Split Time} + 5$

- **Priority order:**

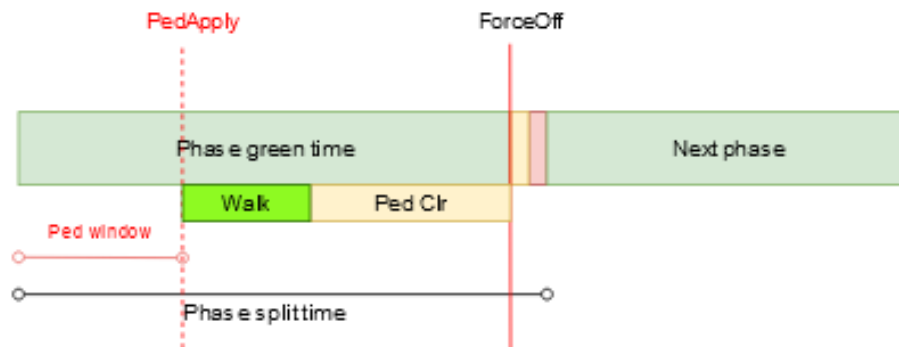
$\text{MinPerm Ped} > \text{Rest-in-Walk} > \text{Stop-in-Walk}$

- **If Stop-in-Walk is OFF**

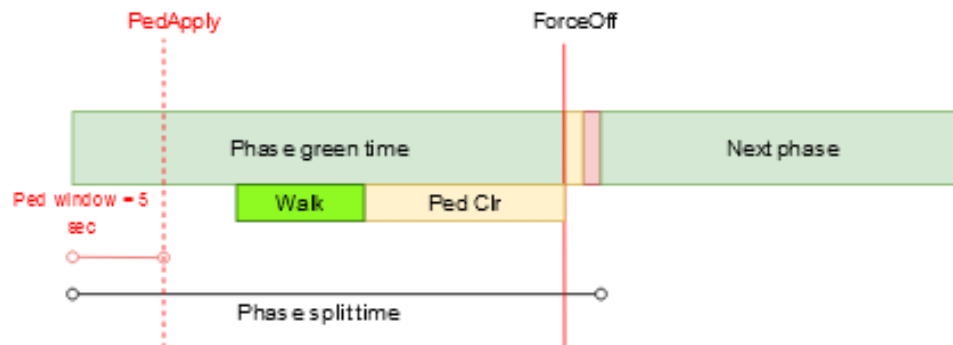
$\text{PedApply} = \text{PedApply} + 1.$

**NOTE: One second is added to the PedApply, calculation to ensure that PedApply cannot begin at the start of the phase green and thus avoid the Ped calls being locked out and not served.**

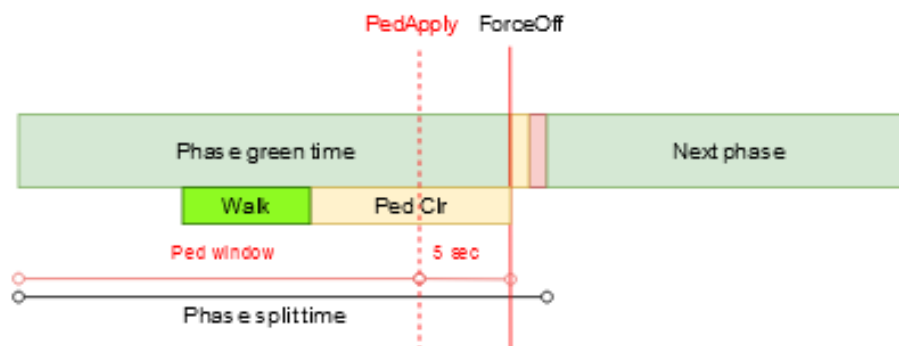
In this coordination enhancement, the calculations of PedApply with *Stop-in-walk or minPerm Ped* setting becomes more **adaptive**. Please see section 1.3 for details on the other modifications made for V85.2 or later.



(1) PedApply Calculation in normal case



(2) PedApply calculation with MinPerm P



(3) PedApply calculation with Stop-in-Walk

## Ped Call

$\text{PedCall} = \text{PedApply} - \max \{ \text{Yellow} + \text{All-Red} \}$  for all phases

## Veh Apply

- **MinPerm Veh not set**

$\text{VehApply} = \text{PrimaryForceOff} - \text{MinGreen} - \min \{ \text{Yellow} + \text{All-Red} \}$  for all phases  
+ 1

- **MinPerm Veh set**

$\text{VehApply} = \text{PrimaryForceOff} - [ \text{split time} - (\text{Yellow} + \text{All-Red}) ]$

- **Walk2 is set w/o Stop-in-Walk**

$\text{VehApply} = \text{PedCall}$ .

### 1.2.3 Transition calculations

The local counter  $\text{Loc}=0$  is referenced to either the beginning (BEGGRN) or end of coord phase green (ENDGRN) by programming the controller offset reference under  $\text{MM}_{>2-}>5$ . This point in the cycle need to line up with the current offset relative to the system time reference (i.e. the Tbc counter subtracts the Programmed offset to insure synchronization across the network.)

The controller calculates the offset error every tick in “SYNC” status or every cycle as below during the transition as shown below:

$$\text{Offset Error} = (\text{Tbc counter} - \text{Prog Offset}) - \text{Local counter}$$

If the correction mode is SHORT/LONG and both short-way & long-way percentages are programmed for the pattern, the controller will calculate the percent of one cycle for long-way/short-way correction, respectively.

$\text{Short-way percentage of one cycle} = \text{cycle time} * \text{short-way percentage} / (\text{short-way percentage} + \text{long-way percentage})$

$\text{Long-way percentage of one cycle} = \text{cycle time} - \text{short-way percent of one cycle};$

If the local counter **lags** the Tbc counter short of the programmed offset, the controller will go to long-way transition. The algorithm will continue to check to verify that the offset error is **larger** than the long-way percentage of one cycle. If it is then the transition will switch to short-way correction.

If the local counter **does not lag** the Tbc counter short of the programmed offset, the controller will go to short-way transition. It will continue to check if the offset error is larger than the short-way percentage of one cycle. If it is then the transition will switch to long-way correction.

During the transition, the controller keeps checking the offset error every single cycle and may switch between long-way and short-way options.

- **Long-way Transition**

The long-way percentage that can be extended in the split time is programmed for the patterns in MM->2->5. Split time is increased using the formula.

$$\text{Long-way Split} = \text{Split} * (100 + \text{Long-way}) / 100$$

The split time enlargement is implemented in the code by increasing the amount of clock ticks of one count. During long-way transition, the time duration of one local count is longer than that of one TBC count.

- **Short-way Transition**

The short-way percentage that can be reduced in the split time is programmed for the patterns in MM->2->5. Split time is reduced using the following formula

$$\text{Short-way Split} = \text{Split} * (100 - \text{Short-way}) / 100$$

The split time deduction is implemented in the code by decreasing the amount of clock ticks of one count. During short-way transition, the time duration of one local count is shorter than that of one TBC count.

- **Dynamic Short-way**

The calculation of the short-way percentage in Dynamic Short-way correction consists of two steps.

1. For each phase in the split table, the software calculates the minimum split time that can accommodate the minimum pedestrian time and green phase time. If the amount of time that the phase used in the last cycle is **larger** than the minimum split time, used the time that the last cycle used. From the above calculated split time, obtain the largest allowable short-way percentage of each phase. If the resultant percentage is smaller than the programmed short-way percentage for the pattern, then take the programmed short-way percentage.
2. Among all active (**currently ON**) phases in all rings, choose the smallest short-way percentage as the current short-way percentage in the transition.

## 1.3 Coordination Modifications in controllers with V85.2 or Later

### 1.3.1 Permissive window Modifications

#### Modification of the PedApply Calculations

- Stop-in-Walk is set in MM->2->1 and is applied to all patterns. For some patterns that have phases whose split times that are long enough to satisfy the minimum pedestrian time, their PedApply points may be unnecessarily late.
- Setting Stop-in-Walk, the PedApply point is calculated to occur 5 sec prior to the force-off point which will accommodate late ped calls. However, calculating in this manner is not necessary for all phases.

- For the phases **without** having enough split time to satisfy the minimum pedestrian requirement, the software can set the PedApply point is set to 5 sec prior to the force-off to accommodate late ped calls because the split time is short.
- For the phases **with** enough split time to satisfy the minimum pedestrian requirement, the calculations are as the following.
  - If split time is at least 5 seconds more than the minimum pedestrian requirement, PedApply is set as the normal:

$$\text{PedApply} = \text{PrimaryForceOff} - (\text{Walk} + \text{PedClearance});$$

- Otherwise, the PedApply is set as the Min P:

$$\text{PedApply} = \text{PrimaryForceOff} + \text{Yellow} + \text{All-Red} - \text{Split Time} + 5.$$

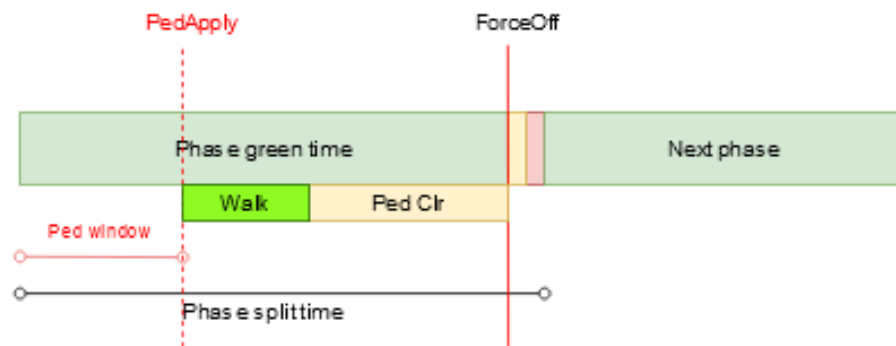
- In similar way, with the Min P setting turned on, the PedApply can be calculated in an adaptive way.
  - If split time is at least 5 seconds more than the minimum pedestrian requirement, the PedApply is set the normal way:

$$\text{PedApply} = \text{PrimaryForceOff} - (\text{Walk} + \text{PedClearance});$$

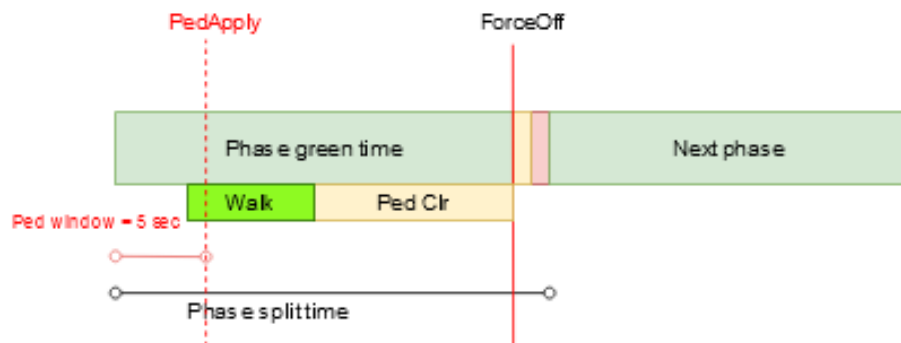
- Otherwise, PedApply is set as the Min P:

$$\text{PedApply} = \text{PrimaryForceOff} + \text{Yellow} + \text{All-Red} - \text{Split Time} + 5.$$

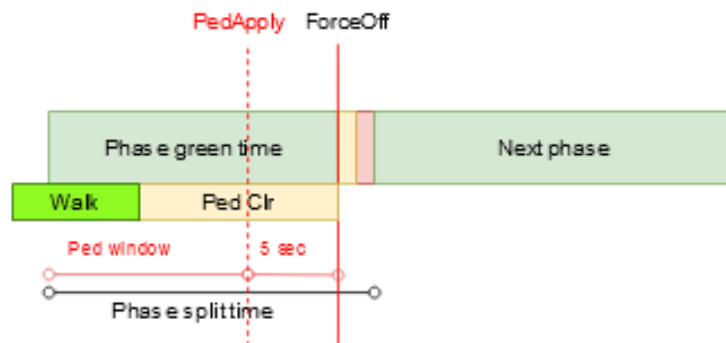
The following picture illustrates the modification ideas of PedApply with Stop-in-Walk. The main purpose of these modification is to reduce unnecessary occurrence of stopping counting at Force-off due to ped timing but also providing no less than 5 sec Ped permissive window.



(1) Stop-in-Walk with the phase time larger than  $W+Pclr+5$



(2) Stop-in-Walk with the phase time larger than  $W+Pclr$  but less than  $W+Pclr+5$



(3) Stop-in-Walk with the phase time less than  $W+Pclr$

### 1.3.2 Transition Modifications

#### Offset estimation

In the controller software, the estimated Offset is calculated using the formula

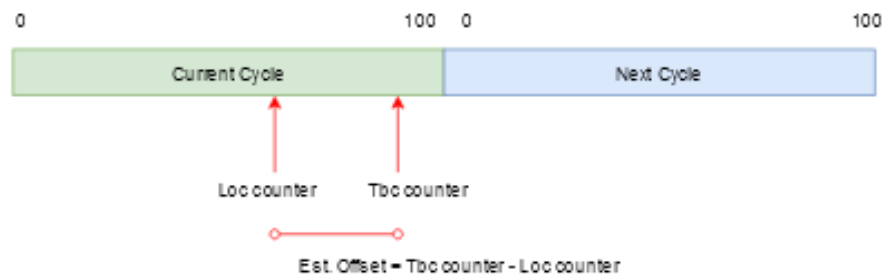
$$\text{Est. Offset} = \text{Tbc counter} - \text{Local counter}$$

if  $\text{Est. Offset} < 0$ ,

$$\text{Est. Offset} = \text{Est. Offset} + \text{Cycle time.}$$

Since both the LOC and TBC counters run in a cycle period, then there are two lengths of offsets between them. In order to minimize the impact of the transition and make transition complete early, the shortest offset is chosen. As demonstrated in the following diagrams, the current estimation method is able to obtain the smallest offset between the local counter and the TBC counter for the normal cases as shown in figure (a). However, in some cases as shown in figure (b), the estimated offset utilizes the larger offset while “**Offset**” indicated in figure(b) is actually the shortest distance between two counters.





(a) The calculation of estl. offset in the code



(b) Estl. offset in the code may choose longer offset in some cases

The modification for the offset calculations are as follows:

$$\text{Error1} = [\text{Local counter} - (\text{Tbc counter} - \text{prog. Offset}) + \text{Cycle time}] \% \text{ Cycle time},$$
$$\text{Error2} = [(\text{Tbc counter} - \text{prog. Offset}) - \text{Local counter} + \text{Cycle time}] \% \text{ Cycle time},$$
$$\text{Error1} = \text{Error1} * 100 / \text{long-way percentage},$$
$$\text{Error2} = \text{Error2} * 100 / \text{short-way percentage},$$

If  $\text{Error1} > \text{Error2}$

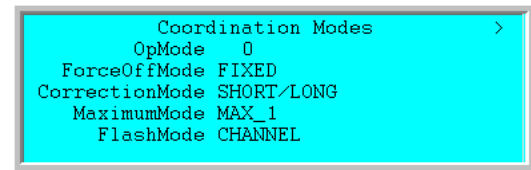
choose short-way transition

otherwise

choose long-way transition

## 1.4 Coordination Modes and Modes plus (MM->2->1)

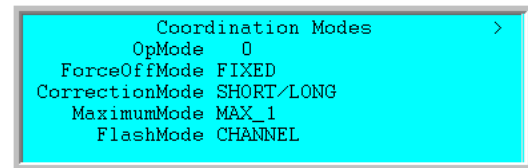
Coordination modes are general modes for all patterns. As such, once set, all patterns adhere to them. Below is a list of features under coordination modes that the user may program. Further details can be found in Chapter 6 of the V85.x/Scout NTCIP controller manuals. Please note that the screens referenced in this chapter are from the V85/Scout controller software using Classic view.



### 1.4.1 Correction Mode

The *Correction Mode* parameter controls whether *Long-way (LONG)* or a combination of *Short-way/Long-way (SHORT/LONG)* transition is used to synchronize offsets during coordination. In addition, the user can also program a dwell in artery transition method, if desired. It is selected on a pattern by pattern basis through the short-way, long-way and dwell settings in the *Trans.CoorØ+* menu (MM->2->5). **DynShortway** (is a feature that will dynamically modify the short-way transition timer to force the controller in-step quicker.

**Considerations:** If you use **Stop-in-walk**, then the **SHORT/LONG** correction mode (with enough percentage of short way transition is allocated in each split) should be selected to get in step during the current cycle thus avoiding a transitions. If available, **DynShortway** can assist in avoiding transitions.



### 1.4.2 Force-Off Mode

Force-offs are predefined points in the signal cycle used to terminate the active phase and limit the time allocated to each active phase.

- |              |  |
|--------------|--|
| <b>FLOAT</b> | Phases other than the coordinated phase(s) are active for their assigned split time only. This causes unused split time to revert to the coordinated phase.  |
| <b>FIXED</b> | Phases are forced-off at fixed points in the cycle. This allows unused split time of a phase to revert to the phases served next in the sequence. Uses <b>Maximum Mode</b> for force off point calculations. |
| <b>OTHER</b> | The coordination mode is not specified by NTCIP. See chapter 2 in this manual for further details.   |

**Considerations:** Use FIXED or FLOAT for adherence to NTCIP

### 1.4.3 Maximum Mode (only used during FIXED)

The *Maximum Mode* parameter determines which maximum green time is active, or if maximum green time is inhibited during coordination. Because it is used under FIXED mode, the user can control the non-artery phase greens, if the phase have constant demand.

- |                |  |
|----------------|--|
| <b>MAX_1</b>   | Selecting the MAX_1 mode allows <i>Maximum 1</i> phase timing to terminate a phase when FIXED or OTHER force-off methods are in effect.  |
| <b>MAX_2</b>   | Selecting the MAX_2 mode allows <i>Maximum 2</i> phase timing to terminate a phase when FIXED or OTHER force-off methods are in effect.  |
| <b>MAX_INH</b> | Selecting MAX_INH inhibits <i>Maximum 1</i> and <i>Maximum 2</i> timing from terminating a phase. When MAX_INH is in effect and a max call is placed on a phase, the max timer will decrement to zero (MM->7->1); however, the phase will not terminate under coordination until it is forced-off. |

**Considerations:** Because it is used under FIXED mode, the user can control the non-artery phase greens, when the phase(s) have constant demand.

#### 1.4.4 Stop-in Walk

*Stop-In-Walk* is a very important feature that allows the split time of a phase less than the minimum pedestrian requirements (sum of the walk + ped clearance + yellow + all-red clearance).

*Stop-In-Walk* causes the local cycle counter to “stop” during coordination if a force-off is applied to the phase and it is still timing walk or pedestrian clearance. *Stop-in-Walk* **will** be used for any pedestrian phase including the artery.

**OFF** *Stop-in-Walk* OFF forces the user to provide adequate split time to service the walk and ped clearance intervals assigned to the phase. The coordination diagnostic will fail the pattern if the split times do not adequately meet the pedestrian requirements.

**ON** *Stop-in-Walk* ON disables the coord diagnostic that insures that the split time is adequate to service the minimum pedestrian times. The local counter will “STOP” at the force-off, thus “suspending” the local cycle timer until the end of ped clearance. At the end of ped clearance, the local cycle counter will begin incrementing and the coordinator will immediately begin correcting the offset using the short-way transition if the splits have enough time to utilize short way for the pattern.

*Note: Rest-in-Walk* programmed for a coord phase defeats *Stop-in-Walk* and requires that pedestrian times be serviced within the programmed split time.

**Considerations:** *Stop-in-Walk* utilizes short-way offset correction. It depends on the user allocating enough time in each split to compensate for the suspended time to avoid a transition. **This feature should only be used when pedestrian actuations are infrequent.** *Stop-In-Walk* may affect arterial phases that are push button actuated when there is no side road demand. If a late arterial Ped call comes in, the coordinator may utilize *Stop-in Walk* to finish processing the arterial Ped clearance times during the first split, thus correcting during the side road splits. If this is not desired, program the arterial phases as *Rest-in-Walk* and program the **Walk Recycle**, **Leave Walk Before** and **Leave Walk After** parameters as described below.

```
< Coordination Modes+
Force-Off+ RESERVED Easy Float OFF
Closed Loop OFF Auto Err Reset OFF
External OFF NTCIP Yield + 0
Latch Sec Frc OFF Coord Hold 0
Stop-in-Walk OFF -- Leave Walk --
Walk Recycle NO_RECYCLE Before TIMED
FreeOnSeqChg OFF After TIMED
ExtPattern OFF NoAddedInit OFF
DynShortway OFF PedCallInh OFF
SyncPulseTime 0 ExtFailPtrn 0
ExtOnCommFail OFF +
```

```
Plan A # Plan B 0
Plan C 0 Plan D 0
```

#### 1.4.5 Walk Recycle

This parameter is used for controlling arterial phases. The Options under this parameter will take effect only when *Rest-In-Walk* is set for the arterial phase(s). **If *Rest-In-Walk* is not set, this parameter is ignored.** When *Rest-In-Walk* is not set, the arterial pedestrians are subject to *PedLeav* and *Ped Yld* parameters as well as opposing phase demand.

*Walk Recycle* and the two *Leave Walk* settings described below, determine how walk intervals are terminated **and recycled** during coordination when the controller is resting in a phase and there is time available to re-service the pedestrian movement before the phase is forced off.

***Walk Recycle* only recycles the walk interval if a ped call has been placed on the phase or if the phase is programmed for *Rest-In-Walk*. A ped recall set through the phase options or through the *Split Table***

```
< Coordination Modes+
Force-Off+ RESERVED Easy Float OFF
Closed Loop OFF Auto Err Reset OFF
External OFF NTCIP Yield + 0
Latch Sec Frc OFF Coord Hold 0
Stop-in-Walk OFF -- Leave Walk --
Walk Recycle NO_RECYCLE Before TIMED
FreeOnSeqChg OFF After TIMED
ExtPattern OFF NoAddedInit OFF
DynShortway OFF PedCallInh OFF
SyncPulseTime 0 ExtFailPtrn 0
ExtOnCommFail OFF +
```

**Mode setting (PED or MxP) will not recycle the walk unless a ped detector has also called the phase or Rest-In-Walk is set.** If you want to rest-in-walk on the arterial phases, then program *Rest-In-Walk* for those phases under menu **MM->1->1->2**.

**NO\_RECYCLE** After servicing walk and ped clearance, the controller will continue to rest in the coordinated phase until the next cycle (Local counter = 0) before deciding to recycle the walk. Walk Recycling is now dependent upon getting a demand from any conflicting phase **AND** a pedestrian actuation or recall on the rest-in-walk phase.

**IMMEDIATE** If *Rest-In-Walk* is set, the controller will recycle the walk immediately (without a pedestrian actuation or recall on the rest-in-walk phase) at the end of ped clearance **if a serviceable (i.e. not inhibited) conflicting call does not exist**. This setting locks out any new conflicting calls until the end of pedestrian clearance in the next cycle. Caution should be used if IMMEDIATE is programmed. One consequence of setting *Walk Recycle* to IMMEDIATE is that side road phases may not be serviced if the recycled ped finishes past the side road phase(s) apply points. There are two ways to solve the above consequence.

If IMMEDIATE recycling is desired, set the *Leave Walk After* parameter to ON DEMAND. This option ignores the PedLeave point and allows the controller to leave walk immediately when a conflicting call is received

Set the *Walk Recycle* parameter to INHIBIT\_1256 or INHIBIT\_3478 as discussed below.

**Ø1256\_INH** This option is useful when the coord phase is Ø4 or Ø8. The coord phase walk is not recycled until the permissive window for the cross street (Ø1256) has had an opportunity to service conflicting pedestrian and vehicle calls.

**Ø3478\_INH** This option is useful when the coord phase is Ø2 or Ø6. The coord phase walk is not recycled until the permissive window for the cross street (Ø3478) has had an opportunity to service conflicting pedestrian and vehicle calls

**NO\_PED\_INH** This option allows the walk of the coord phase to recycle when the pedestrian omits are lifted on the coordinated phase (i.e. the earliest point in the cycle when the coordinator will allow a walk interval to be serviced.) If a ped call is issued during or after ped clearance, the walk will be recycled immediately after the ped clearance is timed and after or at the Ped Yield point of the phase if the controller continues to rest in that phase.

**Considerations:** Walk recycle is a way to achieve reseriving of the artery pedestrians without placing the phase on **PED** or **MxP** recall. This can be helpful when resting in the artery phase with no demand from any other phase. It will use the LEAVE WALK parameters to fully control the arterial Ped recall. It should also be noted that Rest-In-Walk does not operate for uncoordinated phases during short way transitioning.. The **No Short** option (**MM->2->5**) can be turned on, if it desired for Rest-In-Walk to operate for a specific phase, even while in short way transition.

## 1.4.6 Leave Walk Before / Leave Walk After

These parameters apply to the artery phases resting in walk before being recycled. **If *Rest-In-Walk* is not set, this parameter is ignored.**

**Leave Walk Before** applies to the artery phases resting in walk before being recycled. **Leave Walk After** applies to the artery phases resting in walk after the pedestrian is recycled.

The following entries determines when a phase will leave walk if it is resting in walk but has not been recycled:

```
<      Coordination Modes+
Force-Off+ RESERVED    Easy Float OFF
Closed Loop OFF        Auto Err Reset OFF
External OFF           NTCIP Yield + 0
Latch Sec Frc OFF      Coord Hold 0
Stop-in-Walk OFF       -- Leave Walk --
Walk Recycle NO_RECYCLE Before TIMED
FreeOnSeqChg OFF       After TIMED
ExtPattern OFF          NoAddedInit OFF
DynShortway OFF         PedCallInh OFF
SyncPulseTime 0         ExtFailPtrn 0
ExtOnCommFail OFF      +
```

**TIMED** The *PedLeav* point is the latest point in the cycle that allows the controller to begin Ped clearance and have end it at the force-off of the phase. The TIMED option allows the controller to rest-in-walk until the *PedLeav* point when a conflicting call is received on another phase.

**ON DEMND** This option ignores the *PedLeav* point during coordination and allows the controller to leave walk immediately when a conflicting call is received.

**Considerations:** The user is cautioned to test the specific scenarios of these parameters to insure the proper recycling of the artery Ped. The **Leave Walk** features are tightly coupled with the **Walk Recycle** and **Rest-in-walk** features. For example, the **TIMED** parameter uses the calculated *PedLeav* point in the cycle to terminate the ped but the **ON DEMAND** parameter uses demand from another phase to terminate the ped.

## 1.4.7 NTCIP Yield

The *NTCIP Yield* parameter is expressed as a positive and negative number ( - 15 to +15”) and only works with the coordinated phase.. This parameter is used to adjust the default yield point of the coord phase under NTCIP coordination (FIXED and FLOAT modes). This adjustment is applied to only the coordinated phases, whereas the *Early Yield* (MM->2->5) adjustment is applied to all of the non-coordinated phases.

**Considerations:** The user will typically use the default yield points. For example, the default yield points for ENDGRN coordination is ten seconds for the artery phases and zero seconds for the side road phases.

### 1.4.8 Early Yield (EarlyYld)

The *Early Yield* parameter (0-25 seconds) (**MM->2->5**), modifies the yield calculations under NTCIP coordination (FIXED and FLOAT force-off modes). This adjustment is applied to all the **non-coordinated** phases, where the *Coord Yield* adjustment is applied to the coordinated phases.

**Considerations:** The user will typically use the default yield points. For example, the default yield points for ENDGRN coordination is ten seconds for the artery phases and zero seconds for the side road phases.

### 1.4.9 FreeOnSeqChg

Transitioning from one pattern to another is dependent on many decisions such as cycle length changes, coordination phase changes, split time changes and phase sequence changes. Phase Sequence changes can especially influence a transition. This parameter gives the user flexibility to determine when phase sequence changes will occur during coordination pattern changes.

Turning this parameter to **ON** will briefly (approximately 1 second) force the coordinator to run free when a sequence change occurs thus insuring that the coordinator will reset itself. Setting this parameter to **OFF** will run sequence changes when the coordinator deems it is appropriate.

**Considerations:** This normally should be set to ON.

### 1.4.10 No Added Initial

This Feature allows Added Initial Timing to be disabled whenever coordination is active (i.e. Not Free). Set this parameter to ON if you want Added Initial Timing to be disabled during coordination. Set to OFF if you want to continue to use Added Initial Timing during coordination.

**Considerations:** If you have the added initial feature set you should accommodate the Max initial in your split timing and not use this feature.

### 1.4.11 PedCallInh

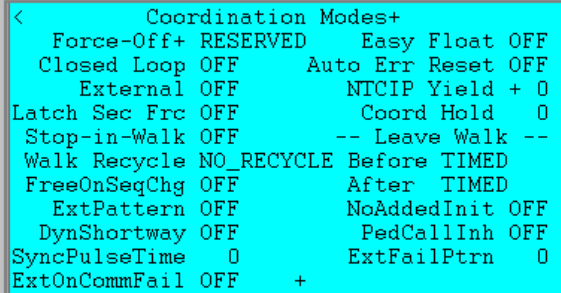
Setting this variable to “ON” will disable pedestrian inhibits during coordination. Thus pedestrians can be recalled.

**Considerations:** Use caution if this is set to “ON” and **Stop-in-Walk** is set.

### 1.4.12 MinPermV/P

These two parameters (found under **MM->2->5**) allow the minimum permissive window for vehicles (V/) and for pedestrians (/P) to be selected on a pattern-by-pattern basis. Enabling this feature **prevents a “late” vehicle and/or pedestrian call from being serviced if the call received after the force-off of the preceding phase.**

**Considerations:** This is useful for STOP-IN\_WALK applications.



Coordination Modes+			
Force-Off+	RESERVED	Easy Float	OFF
Closed Loop	OFF	Auto Err Reset	OFF
External	OFF	NTCIP Yield	+ 0
Latch Sec Frc	OFF	Coord Hold	0
Stop-in-Walk	OFF	-- Leave Walk	--
Walk Recycle	NO_RECYCLE	Before	TIMED
FreeOnSeqChg	OFF	After	TIMED
ExtPattern	OFF	NoAddedInit	OFF
DynShortway	OFF	PedCallInh	OFF
SyncPulseTime	0	ExtFailPtrn	0
ExtOnCommFail	OFF	+	

### 1.4.13 Coord Hold

Coord Hold (0-255) is a length of time in seconds. Coord Hold defines a window immediately following the Force-Off point for the coordinated phase(s) in which the controller may yield to uncoordinated phases if there is serviceable demand. If the coordinated phase(s) is still active once the specified Coord Hold point is reached, a hold will be placed on the coordinated phase(s) until the normal yield point for that phase. This results in the coordinated phase(s) remaining active for the entire cycle. If Coord Hold is programmed, the Artery Ped(s) are allowed to be recycled prior to the Ped apply point if warranted.

As an example, consider a standard eight phase operation using all phases and running phase sequence 1. The user programs Coord Hold as 5 seconds. Phase 2 is the coordinated phase, and coordination is setup for an End of Green reference point. With this programming, phase 2 may yield to phases 3, 4, 5, 6, or 1 only between times 0 to 5 in the Local Cycle.



## 1.5 Easy Calcs Status Screen (MM->2->8->2) for Controllers using Scout/V85.2x or Later

When programming NTCIP coordination, the user has the ability to see what parameters have been calculated by the coordination algorithm. The *Easy Calcs* status screen shows the current force-offs and yield calculations for the active pattern under FIXED, FLOAT or one of the OTHER coordination modes. *Easy Calcs* are identical for the FIXED and FLOAT modes except that “*FloatMx*” is used to limit each non-coordinated phase to its programmed split and move any “slack time” to the coordinated phase. Most users find these default *Easy Calc* calculations acceptable for their application and do not have to review these values with every pattern change.

Easy <>	P	1	2	3	4	5	6	7	8
PrimFrc	65	0	20	45	65	0	20	45	
SecdFrc	65	0	20	45	65	0	20	45	
Veh Yld	0	10	0	0	0	10	0	0	
VehAply	56	91	11	36	56	91	11	36	
Ped Yld	0	10	0	0	0	10	0	0	
PedAply	65	91	20	36	65	91	20	36	
FloatMx	15	30	15	20	15	30	15	20	
PedLeav	65	90	20	35	65	90	20	35	
PedCall	60	85	15	30	60	85	15	30	
SplitRem	0	0	0	0	0	0	0	0	

### 1.5.1 Primary Force-Off

The Primary Force-Off is the point in the local cycle that a force-off is applied to a phase causing that phase to terminate and begin timing yellow clearance. A Primary Force-off will remain applied until the phase terminates.

### 1.5.2 Secondary Force-Off

The Secondary Force-Off is a momentary force-off applied prior to the Primary Force-off. Secondary Force-offs are useful when conditionally servicing phases or when a phase is to be forced off twice per cycle. The Secondary Force-off normally default to the value of Primary Force-off. **NOTE: This feature is not used in NTCIP Coordination.**

### 1.5.3 Vehicle Yield

The Vehicle Yield is that point in the cycle that a vehicle call on a phase will be serviced, i.e. that the phase's inhibit is removed. Note that the phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off.

### 1.5.4 Vehicle Apply

The Vehicle Apply point defines the point in the cycle when the phase inhibit is applied. A phase may begin anytime between the Vehicle Yield point and the Vehicle Apply point.

The Vehicle Apply point (VehAply) for each phase is calculated as:

**Vehicle Apply Point (VehAply) = Primary Force-off – ((Max Yellow + All Red ) + Minimum Green)**

The yield point must be earlier than the automatic application point for the phase to be serviced. If short-cycle offset correction is enabled, the yield point must be earlier still to allow for the effective reduction in split time that occurs when the local cycle timer corrects by running fast.

### 1.5.5 Pedestrian Yield

The Pedestrian Yield is that point in the cycle that a pedestrian call on a phase will be serviced, i.e. that the phases pedestrian inhibit is removed. The phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off.

### 1.5.6 Ped Apply

The Ped Apply point defines the point in the cycle when the pedestrian phase inhibit is applied. A pedestrian phase may begin anytime between the Ped Yield point and the Ped Apply point.

For V85/Scout the PedApply point for each pedestrian phase is calculated as:

Ped Apply Calculation ( all calculation parameters are in seconds)	Modified Parameter (s)
$PedApply = PrimaryForceOff - [ PedClearance + (Walk - 5) \text{ (if Walk > 5)} ] + 1 \text{ ( if PedClearance > 0 )}$	Stop-in Walk=OFF
Note: One possible reason for Walk - 5 would be to make the end of Ped Clearance aligned with the end of Yellow or All-Red because Yellow+All-Red is usually 5 sec	Rest-in Walk = OFF
	Min P (Min Ped Permissive) = OFF
$PedApply = PrimaryForceOff - 5$	Stop-in Walk=ON
$PedApply = PrimaryForceOff - ( Walk + PedClearance ) + 1 \text{ ( if PedClearance > 0 )}$	Rest-in Walk = ON
$PedApply = PrimaryForceOff + 5 - ( Split Time - Yellow - All-Red ) + 1 \text{ ( if PedClearance > 0 )}$	Min P (Min Ped Permissive) = ON
$PedApply = PedApply - 1 - ( Yellow + All-Red )$	Stop-in Walk=OFF and WALK2 is programmed

The same considerations described above for selecting vehicle yield points apply to determining pedestrian yield points except when the STOP-IN-WALK is enabled. Refer to the explanation of Stop-In-Walk.

### 1.5.7 FloatMx

Floating max time (*FloatMx*) is equal to the green portion of the split needed to terminate the phase prior to the force-off if the time allocated to the phase exceeds programmed split time. This is used as the max green time with floating force-offs.

### 1.5.8 PedLeav

The Pedestrian Leave Point is used when Rest-In-Walk is active. This is the point in time when the Pedestrian Clearance begins after the phase has been resting in walk.

### 1.5.9 PedCall

Ped Call displays the last time a call can be placed in the cycle so a pedestrian can be serviced in that cycle. Ped Call is only used when MinP is active, otherwise Ped Call = Ped Apply. The Ped Call point for each pedestrian phase is calculated as:

$$PedCall = Ped Apply - Max (red+yellow)$$

### 1.5.10 SplitRem

This is the remaining time in the split before the next cycle begins.

**Considerations:** This screen is a great diagnostic tool in ascertaining what the software is using to process the phases within a cycle. Once noted there is the **Perm,Frc** screen at **MM->2->7->2** that will allow the user to override the calculated settings. **Keep in Mind that whenever the user changes any coordination parameter that the Easy Calcs may be affected.**

## 1.6 Easy Calcs Status Screen (MM->2->8->2) for Controllers using Scout/V85.1x

When programming NTCIP coordination, the user has the ability to see what parameters have been calculated by the coordination algorithm. The *Easy Calcs* status screen shows the current force-offs and yield calculations for the active pattern under FIXED, FLOAT or one of the OTHER coordination modes. *Easy Calcs* are identical for the FIXED and FLOAT modes except that “*FloatMx*” is used to limit each non-coordinated phase to its programmed split and move any “slack time” to the coordinated phase. Most users find these default *Easy Calc* calculations acceptable for their application and do not have to review these values with every pattern change.

Easy <>	P	1	2	3	4	5	6	7	8
PrimFrc	65	0	20	45	65	0	20	45	
SecdFrc	65	0	20	45	65	0	20	45	
Veh Yld	0	10	0	0	0	10	0	0	
VehAply	56	91	11	36	56	91	11	36	
Ped Yld	0	10	0	0	0	10	0	0	
PedAply	65	91	20	36	65	91	20	36	
FloatMx	15	30	15	20	15	30	15	20	
PedLeav	65	90	20	35	65	90	20	35	
PedCall	60	85	15	30	60	85	15	30	
SplitRem	0	0	0	0	0	0	0	0	

### 1.6.1 Primary Force-Off

The Primary Force-Off is the point in the local cycle that a force-off is applied to a phase causing that phase to terminate and begin timing yellow clearance. A Primary Force-off will remain applied until the phase terminates.

### 1.6.2 Secondary Force-Off

The Secondary Force-Off is a momentary force-off applied prior to the Primary Force-off. Secondary Force-offs are useful when conditionally servicing phases or when a phase is to be forced off twice per cycle. The Secondary Force-off normally default to the value of Primary Force-off. **NOTE: This feature is not used in NTCIP Coordination.**

### 1.6.3 Vehicle Yield

The Vehicle Yield is that point in the cycle that a vehicle call on a phase will be serviced, i.e. that the phase's inhibit is removed. Note that the phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off.

### 1.6.4 Vehicle Apply

The Vehicle Apply point defines the point in the cycle when the phase inhibit is applied. A phase may begin anytime between the Vehicle Yield point and the Vehicle Apply point.

The Vehicle Apply point (VehAply) for each phase is calculated as:

**Vehicle Apply Point (VehAply) = Primary Force-off – ((Max Yellow + All Red ) + Minimum Green)**

The yield point must be earlier than the automatic application point for the phase to be serviced. If short-cycle offset correction is enabled, the yield point must be earlier still to allow for the effective reduction in split time that occurs when the local cycle timer corrects by running fast.

### 1.6.5 Pedestrian Yield

The Pedestrian Yield is that point in the cycle that a pedestrian call on a phase will be serviced, i.e. that the phases pedestrian inhibit is removed. The phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off.

### 1.6.6 Ped Apply

The Ped Apply point defines the point in the cycle when the pedestrian phase inhibit is applied. A pedestrian phase may begin anytime between the Ped Yield point and the Ped Apply point.

For V85/Scout the PedApply point for each pedestrian phase is calculated as:

Ped Apply Calculation ( all calculation parameters are in seconds)	Modified Parameter (s)
$\text{PedApply} = \text{PrimaryForceOff} - [ \text{PedClearance} + (\text{Walk} - 5) \text{ (if Walk > 5)} ] + 1 \text{ (if PedClearance > 0)}$	Stop-in Walk=OFF
Note: One possible reason for Walk - 5 would be to make the end of Ped Clearance aligned with the end of Yellow or All-Red because Yellow+All-Red is usually 5 sec	Rest-in Walk = OFF
	Min P (Min Ped Permissive) = OFF
$\text{PedApply} = \text{PrimaryForceOff} - 5$	Stop-in Walk=ON
$\text{PedApply} = \text{PrimaryForceOff} - ( \text{Walk} + \text{PedClearance} ) + 1 \text{ (if PedClearance > 0)}$	Rest-in Walk = ON
$\text{PedApply} = \text{PrimaryForceOff} + 5 - ( \text{Split Time} - \text{Yellow} - \text{All-Red} ) + 1 \text{ (if PedClearance > 0)}$	Min P (Min Ped Permissive) = ON
$\text{PedApply} = \text{PedApply} - 1 - ( \text{Yellow} + \text{All-Red} )$	Stop-in Walk=OFF and WALK2 is programmed

The same considerations described above for selecting vehicle yield points apply to determining pedestrian yield points except when the STOP-IN-WALK is enabled. Refer to the explanation of Stop-In-Walk.

### 1.6.7 FloatMx

Floating max time (*FloatMx*) is equal to the green portion of the split needed to terminate the phase prior to the force-off if the time allocated to the phase exceeds programmed split time. This is used as the max green time with floating force-offs.

### 1.6.8 PedLeav

The Pedestrian Leave Point is used when Rest-In-Walk is active. This is the point in time when the Pedestrian Clearance begins after the phase has been resting in walk.

### 1.6.9 PedCall

Ped Call displays the last time a call can be placed in the cycle so a pedestrian can be serviced in that cycle. Ped Call is only used when MinP is active, otherwise Ped Call = Ped Apply. The Ped Call point for each pedestrian phase is calculated as:

$$\text{PedCall} = \text{Ped Apply} - \text{Max (red+yellow)}$$

### 1.6.10 SplitRem

This is the remaining time in the split before the next cycle begins.

**Considerations:** This screen is a great diagnostic tool in ascertaining what the software is using to process the phases within a cycle. Once noted there is the **Perm,Frc** screen at **MM->2->7->2** that will allow the user to override the calculated settings. **Keep in Mind that whenever the user changes any coordination parameter that the Easy Calcs may be affected.**

# 1.7 Transition Considerations using NTCIP settings

## 1.7.1 Defining transition

The controller calculates the Time Based Coordination (**TBC**) time based on the cycle length and offset of the current pattern. The Local controller time (**LTBC**) indicates the point in the cycle for a controller to be in 'Sync'. When in 'Sync', these two times will be equal. If these times are not equal the controller is in transition, thus running the local clock faster or shorter so the LTBC will match the TBC time

Coordination transitions occur for varied reasons. They include:

- 1) Changing patterns which will adjust the following coordination parameters:
  - a. Cycle length, Offset or Sequence programming (**MM->2->4**).
  - b. Selecting an Alternate time table and/or a Phase Option table (**MM->2->6**),
  - c. Selecting phase enables or omits and/or coordination phase changes (**MM->2->7->1**).
  - d. Different reference point (selected by the coordinated phase (**MM->2->7->1**) and the **BegGrn / EndGrn** offset reference selection (**MM->2->5**))
  - e. Patterns may be changed by TBC scheduling in the controller, System master pattern selection, or Central commands from ATMS.now.
- 2) If **STOP-IN WALK** (**MM->2->1**) is set to **ON**, then pedestrian calls on phases where splits are less than the sum of the **Walk+ Ped Clearance + Yellow + All Red** times.
- 3) Preemption or transit events
- 4) Beginning to cycle colors after coming out of flash

There are 3 types of transitions. Short way, Long way and Dwell. The following questions will assist the user in monitoring the transition event(s).

- 1) When the controller is in Short way transition – what is the controller doing, and how does that appear to the driver?

During short way transition, the local controller clock is running faster than real time. Phase force off points will be reached sooner. As a result, the driver may observe less green time on a split. Programmed times for Min Green, Yellow, Red, as well Walk and Ped Clearance, are timed.
- 2) When the controller is in long way transition - what is the controller doing, and how does that appear to the driver?

During long way transition, the local controller clock is running slower than real time. Phase force off points will be reached later. As a result, the driver may observe more green time on a split. Programmed times for Min Green, Yellow, Red, as well Walk and Ped Clearance, are timed.
- 3) When the controller is in dwell transition - what is the controller doing, and how does that appear to the driver?

During Dwell transition, the user programs a time, in seconds, to dwell in the coordinated phase (once it is reached in the cycle) and its concurrent phase. As a result, the driver will observe more time on the coordinated approach than they would normally see during the normal running of the pattern.

As discussed above, Transitions depend on specific programming for each intersection including specific Pattern, Pattern+ and Mode/Mode+ selections. The controller will follow the programmed settings, so the user should test their settings prior to field implementation.

## 1.7.2 Real Time Status screens to Monitor

To see pattern status use the Coordination status screen at **MM-7-2** or **MM-2-8-1**. This shows the current pattern and next pattern as well as appropriate offset and cycle length info.

```
OpModes.Src-BTBC  Cycle  Ofst  06:47: 5
Sys-  0 Actv-  1  Loc- 65 Actu: 60 ACTIV
Tbc-  1 Next-  1  Tbc- 25 Err:- 40
Ext-  0 Remo-  0 Prog-100 Prog:  0 LONG
Tod-  1 Test-  0      DynOff: +0 25%
Alt:.Opt.Time.Det.CIR Transit:  0
      0      0      0      0
```

The **Easy Calcs** (**MM-2-8-2**) status screen is a dynamic screen that changes whenever a pattern changes or at the top of each cycle (i.e. Local “0”). The Easy Calcs status screen is an important screen to monitor for the calculated coordination parameters specific to the current pattern and cycle as shown below.

```
Easy <> P..1...2...3...4...5...6...7...8
PrimFrc   65   0  20  45  65   0  20  45
SecdFrc   65   0  20  45  65   0  20  45
Veh Yld    0  10   0   0   0  10   0   0
VehAply   56  91  11  36  56  91  11  36
Ped Yld    0  10   0   0   0  10   0   0
PedAply   65  91  20  36  65  91  20  36
FloatMx   15  30  15  20  15  30  15  20
PedLeav   65  90  20  35  65  90  20  35
PedCall   60  85  15  30  60  85  15  30
SpltRem    0   0   0   0   0   0   0   0
```

Pay attention to the calculated Primary Force Offs and the Vehicle Apply points. These will control when you leave a phase (**PrimFrc**) as well as when the last time in the cycle that a call for a phase has to be present for you to serve that phase (**VehAply**). For example, if you are transitioning and the controller is ready to leave a phase to go to the next phase, it has to be done prior to the Apply point in the cycle. If not, the phase has to wait till the next cycle to be served.

### 1.7.3 Specific settings that affect transitions

#### Basic Phase timings (MM1-1-1)

It should be obvious that all coordination will be affected by basic phase timing such as Minimum green, Maximum green, phase clearance times, and pedestrian timings. All calculations will be keyed off of these timings. To assist the user in controlling these timings, alternate phase and timing tables are provided which can be called by pattern. This allows the user to change/modify timings and coordination calculations.

#### Coord Modes/Modes+(MM-2-1)

The **Correction Mode** should be set to **SHORT/LONG** when possible. This allows the software to choose the most efficient way to get back into sync and limit transition times. Setting it to **LONG**, will lengthen phases times to accommodate the transition using the **Longway** percentage (MM-2-5). Dwelling in the artery is also possible. **Dwell** transition is enabled for a pattern if both Short and Long values are set to zero and Dwell is set to 1-99 seconds. The **Dwell** method corrects the offset by resting at the end of the coordinated phase until the desired offset is reached or until the Dwell time expires

Two **Force Off** modes are provided to distribute unused split time to other phases or just the coordinated phase: **Fixed** and **Float**. **Fixed** time allows unused split time to be given to the next split phase and if that phase doesn't use it, it goes to the following phase, etc. until it gets to the Artery phase. **Float** takes all unused time and gives it back to the artery.

Based on the modes, the times are allocated. With **Float**, calculated "Max" times are created for each phase which is each phase split time minus the phase clearance (Yellow + All Red) times. With **Fixed** mode, the user can program the **Maximum mode** to be **MAX1** (run the phase up to the Max1 time), **MAX2** (run the phase up to the Max2 time) or **MAX\_INH** (run the phase up to the calculated Force off time). During transitions, these settings still hold true and may have an effect when changing Cycles lengths or coordination phases.

**Stop-in-Walk** will modify transition behavior because the coordinator may "stop" while it finishes the pedestrian service and then transition to get back in step. The best practice is to keep the **Stop-in-Walk** difference (in seconds) equal to or less than the sum of the programmed **Shortway** percentages (in seconds) for all phases so that the software can get back in step in the current cycle and not transition over multiple cycles.

If the user is changing phase sequences (rotating phases), setting the parameter **FreeOnSeqChg** to ON will allow the controller will change the sequence at a barrier as well as recalculate cycle parameters based on the new cycle length and phase rotation.

**Dynamic Shortway** is an alternative way to vary the **Shortway** percentage (**MM->2->5**) so make the best use of split time in order to speed up transitions.

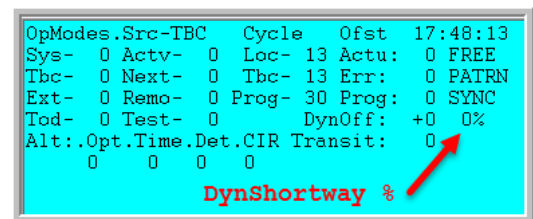
Setting *DynShortway* to **OFF** will use the programmed transition percentage (time).

Setting *DynShortway* to **ON** will result in a **Dynamic Shortway** transition. Then the software does the following:

- 1) It will wait for the controller to be in coordination transition.
- 2) It looks at all the phases that are **ON**
- 3) For **each** phase **ON**, it will calculate the largest **Shortway** percentage that the phase can run and **not** violate its minimums. Note: The controller transition will be based upon the minimum phase times and the amount of time that the phase (split) used in the last cycle.
  - a. It will choose the larger of these values (so, if a phase was skipped, it will choose the min time, else it will use the actual split used).
  - b. If either of these numbers are smaller than the user programmed transition time, the user programmed transition will be used.
- 4) For **all** phases **ON**, it calculates the largest **Shortway** percentage that the phase can run and **not** violate its minimums.
  - a. It chooses the **SMALLEST Shortway** percentage that is calculated for each phase **ON**, because otherwise a larger one would violate the smaller one.
- 5) Once *DynShortway* is set to **ON**, a **Shortway** percentage must be programmed in each pattern. Setting the **Shortway** percentage to a low value such as 1% will allow the algorithm to process.
- 6) Since this is a **Dynamic Shortway** transition, keep in mind that your ability to transition is controlled by which phases are running. Therefore, if a phase that is running that has the standard **Shortway** disabled (i.e. set to "0") or the **Correction Mode** is **LONG**, then obviously no transition will occur. Likewise, if you are running a left turn with a through phase, and the left turn does not have a lot of slop time, then the through phase will be constrained until the left turn terminates.

As an example, assume there is a split that is programmed at 50 seconds. During the last cycle, that split only used 25 seconds. Setting *DynShortway* = **ON** would allow a transition at the speed of 50% during this phase while **not** shortening the time relative to the prior cycle. ( $50\% = ((50 - 25)/50)$ ).

To view *DynShortway* in action, go to the Coord status screen (**MM->2->8->1** or **MM->7->2**).



OpModes.Src-TBC	Cycle	Ofst	17:48:13
Sys- 0 Actv- 0	Loc- 13	Actu: 0	FREE
Tbc- 0 Next- 0	Tbc- 13	Err: 0	PATRN
Ext- 0 Remo- 0	Prog- 30	Prog: 0	SYNC
Tod- 0 Test- 0	DynOff: +0	0%	
Alt:.Opt.Time.Det.CIR	Transit: 0		
0 0 0 0			
<b>DynShortway %</b>			

## Patterns (MM-2-4)

If the user changes pattern lengths or offsets, the software must get in step with the new pattern. For example, if the agency desires that the pattern length change from 90 sec to 120, the controller will need to smooth its way and add 40 seconds to the cycle length. If the agency desires to remain in the artery for until its force off (i.e. setting **Return Hold** to **On** under **MM-2-5**) the new force off may keep the controller in a phase longer than the previous pattern depending on when the transition occurs. Sequence changes will also affect the way the software will get in step with the new pattern. If phase rotations occur, then the controller software will need to get in step with the new rotation as well as recalculate cycle parameters based on the new rotation. It is recommended that the user set the parameter **FreeOnSeqChg** to **ON** so the controller will change the sequence at a barrier as well as recalculate cycle parameters based on the new cycle length and phase rotation.



## 1.8 Perm,Frc (MM->2->7->2)

The Perm,Frc parameters that will allow the user to override the calculated settings for coordination. If a users decides to use this to modify the coordination calculations, he/she should select all settings and change them using the existing Easy Calcs as a base for this change. In addition the user muthe changes to insure the desired operation.

### 1.8.1 Primary Force-Off

The *Primary Force-Off* is the point in the local cycle that a force-off is applied to a phase causing that phase to terminate and begin timing yellow clearance. A *Primary Force-off* will remain applied until the phase terminates. It is up to the user to insure that *Primary Force-Offs* are applied after the minimum phase times of each phase.

The coordination diagnostics does not check minimum phase when force-offs are programmed directly like the FIXED and FLOAT coordination methods. **Therefore, it is possible to program force-offs incorrectly and skip phases.** Care must be taken to ensure that each force-off needs to accommodate the split times including any pedestrians that are programmed. If the phase is skipped for three cycles in a row, the coordinator will fail the pattern. Coord diagnostics provided with FIXED and FLOAT detect these errors before the pattern is run and place the controller in a FREE fail condition.

Spl-	1En<	1...	2...	3...	4...	5...	6...	7...	8>
HoldToMax	.	.	.	.	.	.	.	.	.
OlpedRcl	.	.	.	.	.	.	.	.	.
Mx/Frc	.	0	0	0	0	0	0	0	0
VApply	.	0	0	0	0	0	0	0	0
VehYld	.	0	0	0	0	0	0	0	0
PApply	.	0	0	0	0	0	0	0	0
PedYld	.	0	0	0	0	0	0	0	0

	Beg	End	1234567890123456
Perm1	0	0	.....
Perm2	0	0	.....
Perm3	0	0	.....
	Beg	End	7890123456789012
Perm1	0	0	.....
Perm2	0	0	.....
Perm3	0	0	.....
FrcAll	0		
PedRcy	0		

### 1.8.2 HoldToMax

This parameter will force a phase to be held to its programmed maximum time or the calculated force off if it has a call. This is used for side road phases to ensure that when there is a call, they stay on and are not subject to gapping out.

### 1.8.3 VApply

This parameter allows the user to modify the vehicle apply point for each phase. Please take under consideration when modifying each phases apply point.

The Vehicle Apply point (VehApIy) for each phase is calculated as:

$$\text{Vehicle Apply Point (VehApIy)} = \text{Primary Force-off} - ((\text{Max Yellow} + \text{All Red}) + \text{Minimum Green})$$

The yield point must be earlier than the automatic application point for the phase to be serviced. If short-cycle offset correction is enabled, the yield point must be earlier still to allow for the effective reduction in split time that occurs when the local cycle timer corrects by running fast.

### 1.8.4 VehYld

The *Vehicle Yield* is that point in the cycle that a vehicle call on a phase will be serviced, i.e. that the phase's inhibit is removed. Note that the phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off. The Vehicle Apply point (VehAply) for each phase is calculated as:

$$\text{Vehicle Apply Point (VehAply)} = \text{Primary Force-off} - ((\text{Max Yellow} + \text{All Red}) + \text{Minimum Green})$$

The yield point must be earlier than the automatic application point for the phase to be serviced. If short-cycle offset correction is enabled, the yield point must be earlier still to allow for the effective reduction in split time that occurs when the local cycle timer corrects by running fast.

### 1.8.5 PApply

This parameter allows the user to modify the pedestrian apply point for each phase. Please take under consideration when modifying each phases apply point.

The PedApply point for each pedestrian phase is calculated as per the table below:

Ped Apply Calculation ( all calculation parameters are in seconds)	Modified Parameter (s)
PedApply = PrimaryForceOff - [ PedClearance + (Walk - 5) (if Walk >5) ] + 1 (if PedClearance >0 )	Stop-in Walk=OFF
Note: One possible reason for Walk -5 would be to make the end of Ped Clearance aligned with the end of Yellow or All-Red because Yellow+All-Red is usually 5 sec	Rest-in Walk = OFF
	Min P (Min Ped Permissive) = OFF
PedApply = PrimaryForceOff - 5	Stop-in Walk=ON
PedApply = PrimaryForceOff - ( Walk + PedClearance ) + 1 (if PedClearance >0 )	Rest-in Walk = ON
PedApply = PrimaryForceOff + 5 - ( Split Time - Yellow - All-Red ) + 1 (if PedClearance >0 )	Min P (Min Ped Permissive) = ON
PedApply = PedApply - 1 - ( Yellow + All-Red )	Stop-in Walk=OFF and WALK2 is programmed

The same considerations described above for selecting vehicle yield points apply to determining pedestrian yield points except when the STOP-IN-WALK is enabled. Refer to the explanation of Stop-In-Walk.

### 1.8.6 Pedestrian Yield

The *Pedestrian Yield* is that point in the cycle that a pedestrian call on a phase will be serviced, i.e. that the phases pedestrian inhibit is removed. The phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off per the *PedApply* point which is calculated as shown in the table above.

The same considerations described above for selecting vehicle yield points apply to determining pedestrian yield points except when the STOP-IN-WALK is enabled. Refer to the explanation of Stop-In-Walk.

### 1.8.7 Permissives

The Permissive method allows you to specify up to three permissive “windows of opportunity” to service the yield phases programmed in the *Split Plus Features*. Programming these periods where you allow phases these windows can assist the user in complicated intersections.

### 1.8.8 FrcAll

This is an entry which allows selection of a point along the coordinated cycle that will cause a force-off on any phase which is green. This is programmed in seconds from 0-255.

### 1.8.9 PedRcy

This entry activated when timing the permissive mode in seconds as the point along the coordinated cycle when the coordinated phase(s) recycles to walk

	Beg	End	1234567890123456
Perm1	0	0	.....
Perm2	0	0	.....
Perm3	0	0	.....
	Beg	End	7890123456789012
Perm1	0	0	.....
Perm2	0	0	.....
Perm3	0	0	.....
FrcAll	0		
PedRcy	0		

### 1.8.10 OlpPedRcl

This parameter will place a Ped Recall on any phase that is programmed as a Pedestrian overlap.

### 1.8.11 Mx/Frc

This feature has 3 selection states using the Enable (En) column. The user can toggle between “.”, “F” or “M” for OFF, Primary Force-Off, and Alternate Max.: “F” indicates that the times for each phase are programmed as Primary force off times, “M” indicates that the times programmed for each phase are Alternate Max times and a “.” indicates that this field is not used.

Spl- 1	En<	1	2	3	4	5	6	7	8	>
HoldToMax	.	.	.	.	.	.	.	.	.	.
OlpPedRcl	.	.	.	.	.	.	.	.	.	.
Mx/Frc M	5	23	0	12	16	0	0	0	0	0
VApply	.	0	0	0	0	0	0	0	0	0
VehYld	.	0	0	0	0	0	0	0	0	0
PApply	.	0	0	0	0	0	0	0	0	0
PedYld	.	0	0	0	0	0	0	0	0	0
Beg	End	1234567890123456								
Perm1	0	0	.....							
Perm2	0	0	.....							
Perm3	0	0	.....							

The *Primary Force-Off* is the point in the local cycle that a force-off is applied to a phase causing that phase to terminate and begin timing yellow clearance. A *Primary Force-off* will remain applied until the phase terminates. It is up to the user to ensure if *Primary Force-Offs* are applied after the minimum phase times of each phase.

The coordination diagnostics does not check minimum phase when force-offs are programmed directly like the FIXED and FLOAT coordination methods. **Therefore, it is possible to program force-offs incorrectly and skip phases.** Care must be taken to ensure if each force-off needs to accommodate the split times including any pedestrians that are programmed. If the phase is skipped for three cycles in a row, the coordinator will fail the pattern. Coord diagnostics provided with FIXED and FLOAT detect these errors before the pattern is run and place the controller in a FREE fail condition.

The Alternate Max Times were created to work with Exclusive Pedestrian operations that use split times that are programmed much shorter than the Walk + Ped Clr times. Programming Alternate Max will make it easier to run exclusive pedestrian phases without going into transition. The controller can be programmed with the standard split times setup to accommodate the ped time, and these new Alternate Max times setup as the “alternate split times” which will be used when the exclusive ped phase is skipped. By running fixed force-offs and turning off Max Inhibit in the Coord Modes, each phase maxes out once it reaches its max time rather than distributing all “extra” time to the coordinated phase(s). The new Alternate Max times operate just like any other Max time from Phase Times or the Alternate Phase Times, but this allows you to program just the max green and continue times such as yellow, all-red, min green, etc, from **MM->1->1->1**.

Alternate Max times are programmed in **MM->2->7->2** as shown. **Mx/Frc** can be used to specify a set of Max Green times that should be used when this Split Table is active, or it can be used to modify the Primary Force-Off.

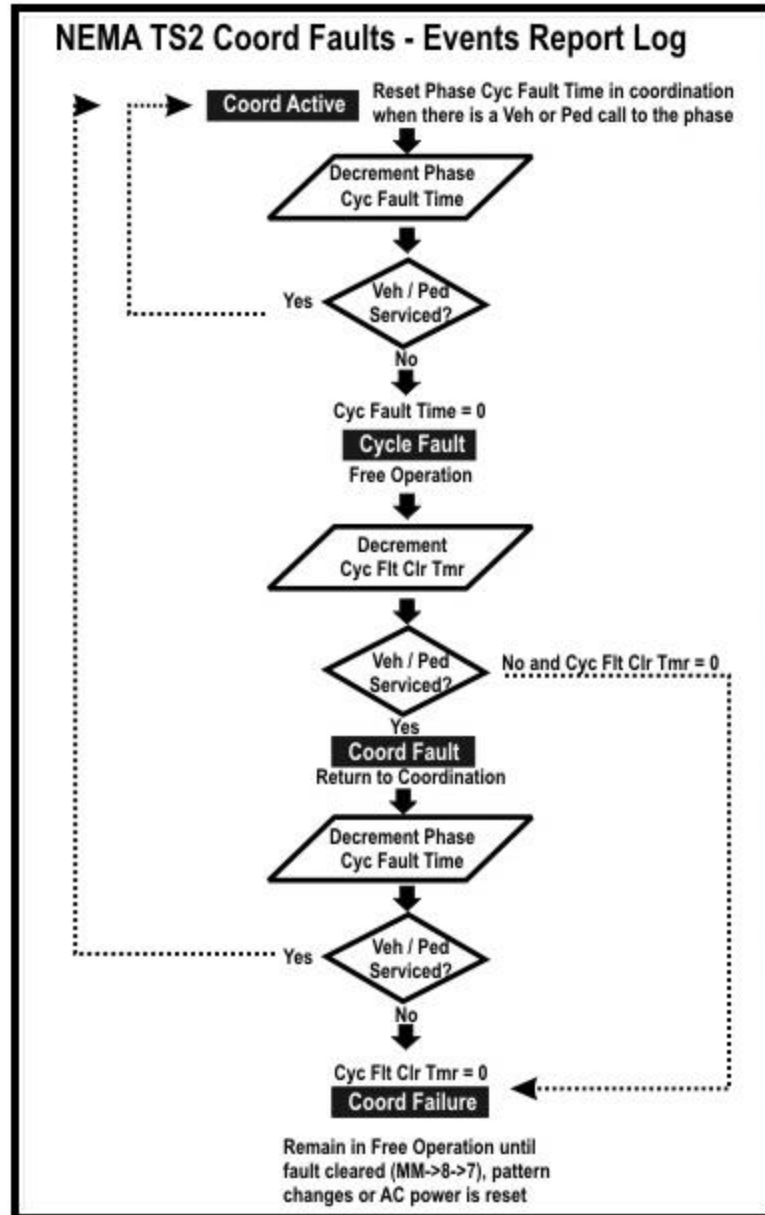
## 1.9 Coordination Alarm Considerations

There are specific alarms that assist the user when programming coordination. They are listed below.

Alarm #	Alarm Name	Description
4	Coordination Failure	This alarm indicates that coordination is failed. There are two ways in which coordination may fail: 1) The TS2 method in which two cycle faults have occurred during coordination, but not when coordination is inactive. 2) A serviceable call has not be serviced in 3 cycles. This is the traditional method, which predates the NEMA TS2 method.
9	Closed Loop Disabled	This alarm, when active, indicates that the Closed-loop Enable parameter is set to OFF.
13	Coordination Free Switch Input	Alarm active when System/Free Switch is FREE
17	Cycle Fault	TS2 Alarm. It indicates that a serviceable call has not been serviced in approximately two cycle times and coordination was active at the time. If the controller is operating in free mode, a Cycle Fault alarm is also logged at the same time as a Cycle Failure alarm.
18	Cycle Failure	TS2 Alarm. It indicates that a serviceable call has not been serviced in approximately two cycle times and that coordination was not active at the time.
19	Coordination Fault	Indicates that a cycle fault occurred during coordination.
30	Pattern Error / Coord Diagnostic Fault	Active when coord diagnostic has failed.
38	Pattern Change	Coordination Pattern changes are logged to the event and alarm buffers using this alarm number. The data byte stores the new pattern number.
47	Coord Active	Set when coordination is active (not free)
60	Coordination Failure	Alarm is ON when Coordination has failed
61	Coordination in (Sync) Transition	Alarm is ON when coord is active and in transition for times over 3 seconds. Alarm is OFF when coord is active and in SYNC.

### 1.9.1 Algorithmic details of various coordination alarms

In particular, Cycle Fault (Alarm #16) and Cycle Failure (Alarm # 17) alarms may occur if the user does not program the coordination parameters correctly. Prior to declaring a specific coordination alarm, the controller software will run as per the following flowchart.

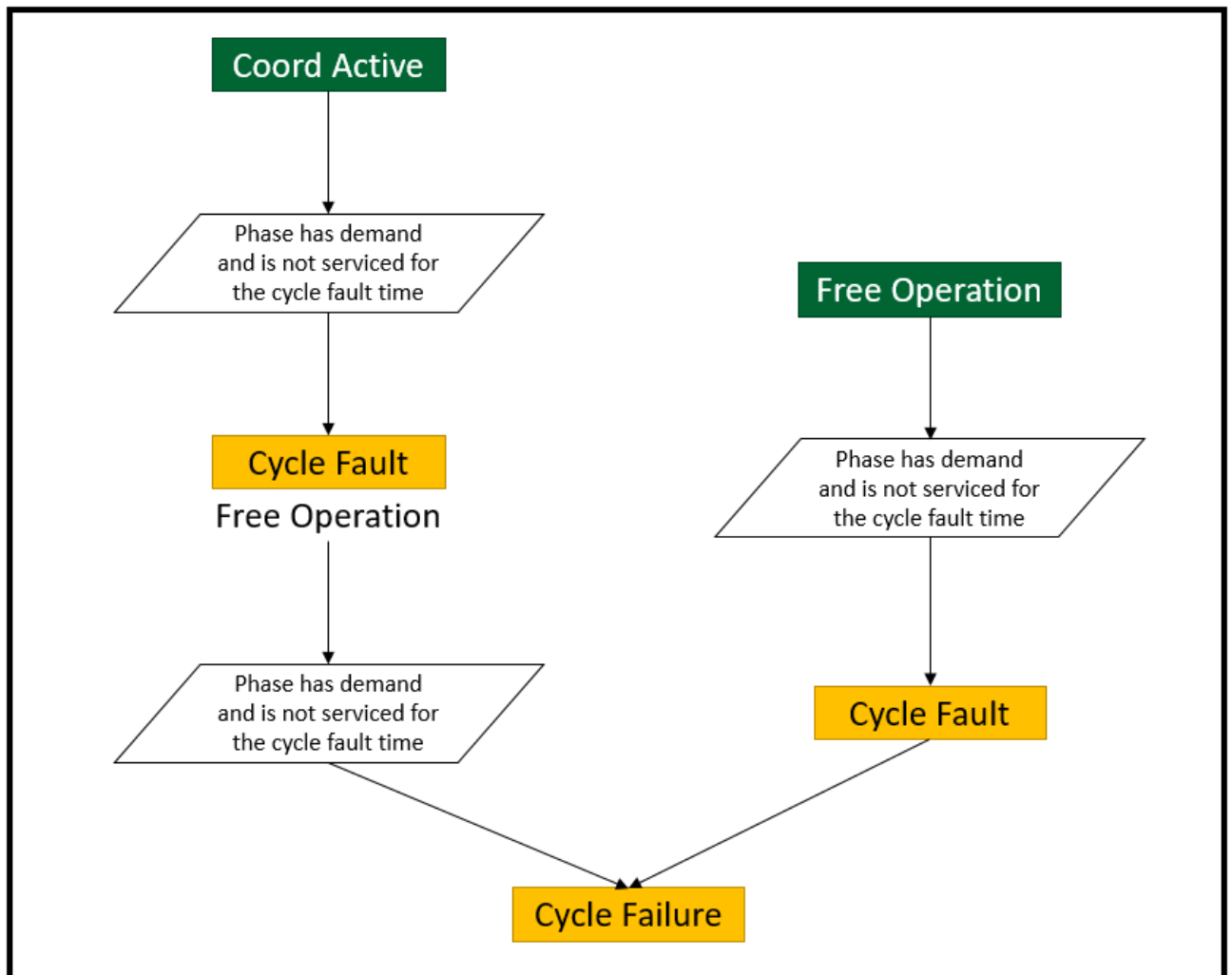


1) The controller software will first establish the amount of time that must expire without a phase being serviced in order to declare a fault (“cycle fault time”). That amount of time is dependent upon a few settings – the phase mode (STD8, USER, etc), whether the controller is in free or coord, and whether or not the user entered a max cycle time in the unit parameters.

Phase Mode	Coord State	Max Cycle Time	Cycle Fault Time
STD8/QSEQ/DIA	Free	0	calculated from maxes
STD8/QSEQ/DIA	Free	>30	user settable time (MM-1-2-1)
STD8/QSEQ/DIA	Coord	n/a	3 x pattern cycle
USER	Free	0	420"
USER	Free	>60	user settable time (MM-1-2-1)
USER	Coord	n/a	3 x pattern cycle

2) Secondly, the controller monitors the phases to see if any phase, that had demand, was not serviced for the cycle fault time. If a fault occurs, the action is based upon user settings as follows:

- In all cases a “*cycle fault*” is declared.
- If the controller is running free then a “*cycle failure*” occurs
- If the controller is running coordination then a “*coord cycle fault*” will occurs on the first occurrence of a cycle fault.
- Once a fault occurs while running coordination, if the fault clears but occurs again before 4x the cycle fault time, then a “*coord cycle fail*” will occur, and the controller will latch in a free state.
- Once a fault occurs for any reason or any amount of times, a timer is set to the cycle fault time. If the timer expires before the fault is cleared, then a “*cycle failure*” will occur. (The user can cause the controller to go to flash in this case). Although the algorithm is programmed for this event, **THIS SHOULD NEVER HAPPEN.**



In particular, below are further details on how the software relates to the coordination alarms.

#### **Alarm #17 Cycle Fault**

Any time a cycle fault occurs during coordination (a phase is not service for the fault timer amount of time) for any reason, the Cycle Fault alarm is set. If it occurs during coordination or preemption the data element of the event will tell you if it was caused during coordination or preempt. If it was during preemption, the data will also tell you which preemption interval. A cycle fault is like a “first time forgiven” skipped phase.

#### **Alarm #18 Cycle Failure**

Any time a cycle fault occurs during free operation, a Cycle Failure alarm occurs. Anytime during coordination that a cycle fault occurred and did not clear for the “*cycle fault clear time*”, a Cycle Failure occurs. Another way to view the Cycle Failure alarm is a way for the software to indicate an issue with the cycle. This failure occurred because it happened during free and/or the coord/preempt fault did not clear itself when the controller went free. A Cycle Failure is a critical coordination alarm that should normally not occur.

#### **Alarm #19 Coord Cycle Fault**

Any time a cycle fault occurs during coordination, the Coord Cycle Fault alarm is set.

#### **Alarm #4 Coord Cycle Failure**

Any time a cycle fault occurs a second time **BEFORE** the “*cycle fault clear time*” expires after the prior cycle fault, a Coord Cycle Failure alarm is set. If you enable this alarm, then the failure is latched, and the controller will stay free until the fault is cleared. If you do not enable this alarm, then the failure is not latched, and the controller will run coordination once the fault is cleared.

The following programming parameters should be considered:

#### **Auto Err Reset (MM->2->1)**

If the auto error reset feature is enabled in the coordination Mode parameters, then this will allow a new pattern to clear a cycle fault that was latched.

#### **Max Cycle Tm (MM-1-2-1)**

*Maximum-Cycle-Time* is a manual override value used to check that the controller is cycling properly. If no value is entered, the controller will calculate a value based on the controller phase and coordination programming as shown in the section above.

#### **Cycle Failure Action (MM-1-2-1)**

As explained above, a cycle failure is considered a critical problem, because it means that a phase was skipped in free or that once coordination went free, the phase that was skipped never ran. The controller gives you the option to report it as an alarm, and keep running – or, send the cabinet into flash.

For emphasis, this should simply never happen. The controller software is **NOT DESIGNED TO SKIP PHASES**. For this reason, the user can send the controller to flash when this does occur.

### 1.9.2 Alarm 17: Cycle Fault

Fault #	Fault Description
0	Other cycle fault
1	Non-preempt cycle fault (not servicing phases)
2	Preempt cycle fault (timed out while seeking <b>track</b> phases)
3	Preempt cycle fault (timed out while seeking <b>dwel</b> phases)
4	4 Preempt cycle fault (timed out while seeking <b>return/end of preempt</b> )

### 1.9.3 Alarm 30 Pattern Error Faults

Fault #	Fault Description
0	No Error
1	In diamond mode, sum of major phases (splits) adds to zero
2	In diamond mode, sum of splits did not equal cycle length
3	Sum of splits exceeded max cycle length (max length currently 999 in ATC/2070, 255 in 980/v65 or older)
4	Invalid split number called out in pattern
5	Ring 1 / 2 sum of splits not equal (when applicable)
6	Split time is shorter than sum of min time for a phase
7	Coordinated phases are not compatible
8	No coordinated phase assigned
9	More than one coord phase was designated for a single ring
10	Undefined
11	Fastway/Shortway transition time greater than 25% (out of range)
12	Undefined
13	Stop-time active
14	Manual control active
15	Error in cycle length when calculating reference point (Cycle time is greater than calculated coord max cycle length)
16	In diamond mode, error in phase split value (typically phase 12)
17	Active split had a zero split value programmed



## 1.10 User mode diagnostics

**Note:** When considering coordination, using the STD8 phase mode will take advantage of the most coordination diagnostic checks to catch common data entry mistakes, and if detected, times the intersection in FREE. In USER mode, most of these coordination diagnostics are removed, and the onus is on the agency verify and test the programming to ensure that coordination pattern(s) run as expected.

Programming an intersection in Standard 8 phase operation has built in coordination diagnostics. When programming an intersection in User mode, diagnostics is limited. The list below should be considered when programming user mode. If you follow these rules of thumb, then coordination should run properly.

- The sum of splits must equal cycle length in all rings.
- Within a barrier, splits in ring 1 must equal ring 2 and equal ring 3, etc.
- There must be a phase in each barrier in each ring that is used.
- The reference point must touch a barrier.
- User testing. The user must:
  - Run all patterns to verify phase checks and easy calcs
  - Run all patterns with max recall to verify correct splits
  - Run all patterns with all ped call if Stop-in-walk is ON to verify if skipping occurs. The user may need to adjust No-short or % Shortway to accommodate transitions caused by walk times.
- The user is ultimately responsible for checking the Easy Calcs and verifying the correct operation at the intersection.

## 1.11 Standard 8 Phase operation verses User mode operation- General considerations

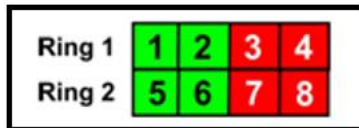
- Reference phases are selected differently. For a given ring, the force-offs/ yields /etc. (things you see on the Easy Calcs screen) are calculated from the reference phase. So, selection of a different reference phase can result in coordination point calculation differences.
- Return hold is applied differently. In STD8, there are a fair number of conditions such as the type of recalls applied as to how return hold is applied. For USER mode, if you are a coordination reference phase, you simply get a return hold.
- Walk Recycle selections (specifically the “1256/3478 inhibit) mode works the same.
- Coordination diagnostic checks are different between STD8 and USER. The USER mode diagnostics is less specific and thus the agency must test the coordination integrity prior to placing the controller in the field. This includes checking and validating that the sequence and concurrency is programmed properly.
- If only one reference phase is programmed, the “pseudo-coordination” phase(s) in other rings are selected differently for STD8 and USER mode. For a given ring, the force-offs/yields/etc. (things you see on the Easy Calcs screen) are calculated based on the reference phase and “pseudo-coordination” phase(s). So, selection of a different reference phase can result in coordination point calculation differences.
- Independent ring offsets can be programmed in USER mode. This feature is available in V85.x/Scout only.

## 1.12 Coordination Examples

The following is an example coordination setup and the changes that occur when modifying the features described in the above sections. This section will use the following coordination setup and make modifications to show the implications to intersection control.

### 1.12.1 Basic program

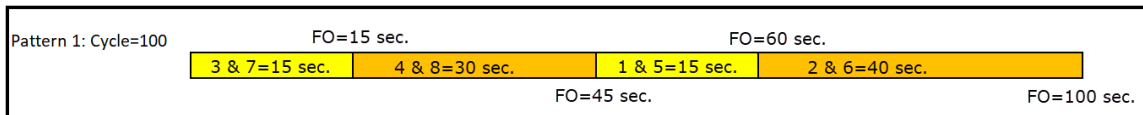
The intersection that will be modified is a standard 8 phase intersection running sequence 1 as shown below:



Each phase will use the following base timing and base coordination setup:

Timing in seconds								
Phase	1	2	3	4	5	6	7	8
Min green	5	20	5	10	5	20	5	10
Gap	1	2	1	2	1	2	1	2
Max 1	10	30	10	20	10	30	10	20
Max 2	15	35	15	25	15	35	15	25
Yellow	3	4	3	4	3	4	3	4
Red	2	2	2	2	2	2	2	2
Walk		10		7		10		7
Ped Clr		10		10		10		10
Pattern 1: Cycle=100								
Splits	15	40	15	30	15	40	15	30

The coordination reference point will be ENDGRN and the coord phase will be Phase 2. The cycle is displayed as shown below:



The basic programming screens using V80.x or V85.x/Scout are shown below

<pre> &lt; Options      ..1..2..3..4..5..6..7..8&gt;   Enable P #   X  X  X  X  X  X  X  X   Min Recall   .  .  .  .  .  .  .  .   Max Recall   .  .  .  .  .  .  .  .   Ped Recall   .  .  .  .  .  .  .  .   Soft Recall  .  .  .  .  .  .  .  .   Lock Calls   .  .  .  .  .  .  .  .   Auto Flash Entry . . . . . . . .   Auto Flash Exit . . . . . . . .   Dual Entry   X  .  X  .  X  .  X   Enable Simul Gap X X X X X X X X   Guarantd Passage . . . . . . . .   Rest In Walk+ . . . . . . . . </pre> <p>MM-&gt;1-&gt;1-&gt;2</p>	<pre> Times &lt; &gt; ..1..2..3..4..5..6..7..8 Min Grn      5  20  5  10  5  20  5  10 Gap,Ext      1.0 2.0 1.0 2.0 1.0 2.0 1.0 2.0 Max 1        10  30  10  20  10  30  10  20 Max 2        15  35  15  25  15  35  15  25 Yel Clr      3.0 4.0 3.0 4.0 3.0 4.0 3.0 4.0 Red Clr      2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 Walk         0  10  0  7  0  10  0  7 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  0  0  0  0  0  0 Gap Reduce </pre> <p>MM-&gt;1-&gt;1-&gt;1</p>
<pre> Coordination Modes &gt;   OpMode      0   ForceOffMode FIXED   CorrectionMode SHORT/LONG   MaximumMode MAX_1   FlashMode   CHANNEL </pre> <p>MM-&gt;2-&gt;1</p>	<pre> &lt; Coordination Modes+   Force-Off+ RESERVED      Easy Float OFF   Closed Loop ON           Auto Err Reset ON   External OFF             NTCIP Yield + 0   Latch Sec Frc OFF        -- Leave Walk --   Stop-in-Walk OFF         Before TIMED   Walk Recycle NO_RECYCLE After TIMED   FreeOnSeqChg ON          NoAddedInit OFF   ExtPattern OFF           PedCallInh OFF   DynShortway ON           ExtFailPtrn 0   SyncPulseTime 0          ExtOnCommFail OFF   Plan A 0                Plan B 0   Plan C 0                Plan D 0 </pre> <p>MM-&gt;2-&gt;1</p>
<pre> Pat#   Cycle   Offset   Split   Seqnc 1      100     0         1         1 2       0       0         0         1 3       0       0         0         1 4       0       0         0         1 5       0       0         0         1 6       0       0         0         1 7       0       0         0         1 8       0       0         0         1 9       0       0         0         1 10      0       0         0         1 11      0       0         0         1 12 +    0       0         0         1 </pre> <p>MM-&gt;2-&gt;4</p>	<pre> Spl- 1&lt;&gt;P..1..2..3..4..5..6..7..8 Time   15  40  15  30  15  40  15  30 Coor-P   X  .  .  .  .  .  .  . Mode     NON MAX NON NON NON MAX NON NON </pre> <p>MM-&gt;2-&gt;7-&gt;1</p>
<pre> Pt Sht Lng Dwl Ely Off Ret Min &gt;   V P % MI 1 10 25 0 0 End X . . . . . 2 0 17 0 0 Beg . . . . . 3 0 17 0 0 Beg . . . . . 4 0 17 0 0 Beg . . . . . 5 0 17 0 0 Beg . . . . . 6 0 17 0 0 Beg . . . . . 7 0 17 0 0 Beg . . . . . 8 0 17 0 0 Beg . . . . . 9 0 17 0 0 Beg . . . . . 10 0 17 0 0 Beg . . . . . 11 + 0 17 0 0 Beg . . . . . </pre> <p>MM-&gt;2-&gt;5</p>	<pre> &lt; No Shortway 1 2 3 Pt 12345678 90123456 78901234 56789012 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6 ..... 7 ..... 8 ..... 9 ..... 10 ..... 11 + ..... </pre> <p>MM-&gt;2-&gt;5</p>

Running this in the controller with constant calls on each phase and peds 2,4,6, and 8, the following is the calculated values shown under **MM->2->8->2**, Easy calcs.

Easy <>	P..1...2...3...4...5...6...7...8
PrimFrc	61 0 16 45 61 0 16 45
SecdFrc	61 0 16 45 61 0 16 45
Veh Yld	0 10 0 0 0 10 0 0
VehAppl	52 76 7 31 52 76 7 31
Ped Yld	0 10 0 0 0 10 0 0
PedAppl	61 86 16 34 61 86 16 34
FloatMx	10 34 10 24 10 34 10 24
PedLeav	61 90 16 35 61 90 16 35
PedCall	55 79 10 27 55 79 10 27
SpltRem	0 98 0 0 0 98 0 0

### 1.12.2 Ped Recycle Considerations

Now let's set up the artery peds to be recycled. Please be reminded that all other phases require calls to be serviced. First under **MM->1->1->2** we will set Rest-in-walk on Phases 2 & 6. Under **MM->2->1**, set Walk Recycle as No Recycle and the and the Leave Walk values as Timed.

< Options	..1..2...3...4...5...6...7...8>
Enable P X	X X X X X X X X
Min Recall .	. . . . . . . .
Max Recall .	. . . . . . . .
Ped Recall .	. . . . . . . .
Soft Recall .	. . . . . . . .
Lock Calls .	. . . . . . . .
Auto Flash Entry	. . . . . . . .
Auto Flash Exit	. . . . . . . .
Dual Entry .	X . X . X . X .
Enable Simul Gap	X X X X X X X X
Guarantd Passage	. . . . . . . .
Rest In Walk+	X . . . X . . .

<	Coordination Modes+
Force-Off+	RESERVED Easy Float OFF
Closed Loop	ON Auto Err Reset ON
External	OFF NTCIP Yield + 0
Latch Sec Frc	OFF -- Leave walk --
Stop-in-Walk	OFF Before TIMED
Walk Recycle	NO_RECYCLE After TIMED
FreeonSeqChg	ON NoAddedInit OFF
ExtPattern	OFF PedCallInh OFF
DynShortway	ON ExtFailPtrn 0
SyncPulseTime	0 ExtOnCommFail OFF
Plan A	0 Plan B 0
Plan C	0 Plan D 0

The Easy calcs will change to:

Easy <>	P..1...2...3...4...5...6...7...8
PrimFrc	61 0 16 45 61 0 16 45
SecdFrc	61 0 16 45 61 0 16 45
Veh Yld	0 10 0 0 0 10 0 0
VehAppl	52 76 7 31 52 76 7 31
Ped Yld	0 10 0 0 0 10 0 0
PedAppl	61 81 16 34 61 81 16 34
FloatMx	10 34 10 24 10 34 10 24
PedLeav	61 90 16 35 61 90 16 35
PedCall	55 79 10 27 55 79 10 27
SpltRem	0 98 0 0 0 98 0 0

Original calcs

Easy <>	P..1...2...3...4...5...6...7...8
PrimFrc	61 0 16 45 61 0 16 45
SecdFrc	61 0 16 45 61 0 16 45
Veh Yld	0 10 0 0 0 10 0 0
VehAppl	52 76 7 31 52 76 7 31
Ped Yld	0 10 0 0 0 10 0 0
PedAppl	61 81 16 34 61 81 16 34
FloatMx	10 34 10 24 10 34 10 24
PedLeav	61 90 16 35 61 90 16 35
PedCall	55 74 10 27 55 74 10 27
SpltRem	0 8 0 0 0 8 0 0

Modified Calcs

Notice that the ped apply point was reduced to insure the ped will not be recycled before moving onto the next cycle. If you change the Walk Recycle to P\_3478\_INH, IMMEDIATE or NO\_PED\_INH. Notice that the modifications are the same as above

Easy <>	P..1...2...3...4...5...6...7...8
PrimFrc	61 0 16 45 61 0 16 45
SecdFrc	61 0 16 45 61 0 16 45
Veh Yld	0 10 0 0 0 10 0 0
VehAppl	52 76 7 31 52 76 7 31
Ped Yld	0 10 0 0 0 10 0 0
PedAppl	61 81 16 34 61 81 16 34
FloatMx	10 34 10 24 10 34 10 24
PedLeav	61 90 16 35 61 90 16 35
PedCall	55 79 10 27 55 79 10 27
SpltRem	0 98 0 0 0 98 0 0

Original calcs

Easy <>	P..1...2...3...4...5...6...7...8
PrimFrc	61 0 16 45 61 0 16 45
SecdFrc	61 0 16 45 61 0 16 45
Veh Yld	0 10 0 0 0 10 0 0
VehAppl	52 76 7 31 52 76 7 31
Ped Yld	0 10 0 0 0 10 0 0
PedAppl	61 81 16 34 61 81 16 34
FloatMx	10 34 10 24 10 34 10 24
PedLeav	61 90 16 35 61 90 16 35
PedCall	55 74 10 27 55 74 10 27
SpltRem	0 0 0 7 0 0 0 7

Modified Calcs

Now change the Leave walk parameters to ON DEMND and select Walk Recycle as IMMEDIATE or NO\_PED\_INH.

< Coordination Modes+			
Force-Off+	RESERVED	Easy Float	OFF
Closed Loop	ON	Auto Err Reset	ON
External	OFF	NTCIP Yield	+ 0
Latch Sec Frc	OFF	-- Leave Walk	-
Stop-in-Walk	OFF	Before ON DEMND	
Walk Recycle	IMMEDIATE	After ON DEMND	
FreeOnSeqChg	ON	NoAddedInit	OFF
ExtPattern	OFF	PedCallInh	OFF
DynShortway	ON	ExtFailPtrn	0
SyncPulseTime	0	ExtOnCommFail	OFF
Plan A	0	Plan B	0
Plan C	0	Plan D	0

or

< Coordination Modes+			
Force-Off+	RESERVED	Easy Float	OFF
Closed Loop	ON	Auto Err Reset	ON
External	OFF	NTCIP Yield	+ 0
Latch Sec Frc	OFF	-- Leave Walk	-
Stop-in-Walk	OFF	Before ON DEMND	
Walk Recycle	NO_PED_INH	After ON DEMND	
FreeOnSeqChg	ON	NoAddedInit	OFF
ExtPattern	OFF	PedCallInh	OFF
DynShortway	ON	ExtFailPtrn	0
SyncPulseTime	0	ExtOnCommFail	OFF
Plan A	0	Plan B	0
Plan C	0	Plan D	0

The easy calcs are calculated as shown below.

Easy <> P..1...2...3...4...5...6...7...8		Easy <> P..1...2...3...4...5...6...7...8	
PrimFrc	61 0 16 45 61 0 16 45	PrimFrc	61 0 16 45 61 0 16 45
SecdFrc	61 0 16 45 61 0 16 45	SecdFrc	61 0 16 45 61 0 16 45
Veh Yld	0 10 0 0 0 10 0 0	Veh Yld	0 10 0 0 0 10 0 0
VehAppl	52 76 7 31 52 76 7 31	VehAppl	52 76 7 31 52 76 7 31
Ped Yld	0 10 0 0 0 10 0 0	Ped Yld	0 10 0 0 0 10 0 0
PedAppl	61 86 16 34 61 86 16 34	PedAppl	61 81 16 34 61 81 16 34
FloatMx	10 34 10 24 10 34 10 24	FloatMx	10 34 10 24 10 34 10 24
PedLeav	61 90 16 35 61 90 16 35	PedLeav	61 90 16 35 61 90 16 35
PedCall	55 79 10 27 55 79 10 27	PedCall	55 74 10 27 55 74 10 27
SpltRem	0 98 0 0 0 98 0 0	SpltRem	0 0 0 7 0 0 0 7

Original calcs

Modified Calcs

However, when running the controller, you will notice that phases 3 & 7 will be skipped 3 cycles in a row and the coordinator will fail. Why??? Recall the definition of ON DEMND for the Leave Walk parameter which is **“This option ignores the PedLeav point during coordination and allows the controller to leave walk immediately when a conflicting call is received.”** Because the Ped Leave point is ignored, and Rest-in-Walk is on, we will start the Ped Clearance at the Force off point, which is the zero point of the next cycle. Running a Ped clearance of 10 seconds will pass the apply point of phases 3 & 7 which will “skip” them. One way to mitigate this is to set a ped recall on other phase. Another way is to set the Leave Walk parameters set to TIMED.

### 1.12.3 Stop-in walk considerations.

Using the same intersection, let’s consider the effects of using Stop-in walk. The basic programming screens using V80.x or V85.x/Scout are shown below. Note that Stop in Walk is set and the split times for phases are different. We have purposely changed the split time to be less than the time needed to run the ped and clear the phases for phases 4 and 8. In this way stop in walk occur when Phases 4 or 8 are called.

< Options ..1..2...3...4...5...6...7...8>		Times < >..1...2...3...4...5...6...7...8	
Enable P #	X X X X X X X X	Min Grn	5 20 5 10 5 20 5 10
Min Recall	. . . . .	Gap,Ext	1.0 2.0 1.0 2.0 1.0 2.0 1.0 2.0
Max Recall	. . . . .	Max 1	10 30 10 20 10 30 10 20
Ped Recall	. . . . .	Max 2	15 35 15 25 15 35 15 25
Soft Recall	. . . . .	Yel Clr	3.0 4.0 3.0 4.0 3.0 4.0 3.0 4.0
Lock Calls	. . . . .	Red Clr	2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0
Auto Flash Entry	. . . . .	Walk	0 10 0 7 0 10 0 7
Auto Flash Exit	. . . . .	Ped Clr	0 10 0 10 0 10 0 10
Dual Entry	X . X . X . X . X	Red Revl	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Enable Simul Gap	X X X X X X X X	Add Init	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Guarantd Passage	. . . . .	Max Init	0 0 0 0 0 0 0 0
Rest In Walk+	. . . . .	Gap Reduce	

MM->1->1->2

MM->1->1->1

```

Coordination Modes >
OpMode 0
ForceOffMode FIXED
CorrectionMode SHORT/LONG
MaximumMode MAX_1
FlashMode CHANNEL

```

MM->2->1

```

< Coordination Modes+
Force-Off+ RESERVED Easy Float OFF
Closed Loop ON Auto Err Reset ON
External OFF NTCIP Yield + 0
Latch_Sec_Err OFF -- Leave Walk --
Stop-in-Walk ON Before TIMED
Walk Recycle NO_RECYCLE After TIMED
FreeOnSeqChg ON NoAddedInit OFF
ExtPattern OFF PedCallInh OFF
DynShortway ON ExtFailPtrn 0
SyncPulseTime 0 ExtOnCommFail OFF
Plan A 0 Plan B 0
Plan C 0 Plan D 0

```

MM->2->1

0

```

Pat# Cycle Offset Split Seqnc
1 100 0 1 1
2 0 0 0 1
3 0 0 0 1
4 0 0 0 1
5 0 0 0 1
6 0 0 0 1
7 0 0 0 1
8 0 0 0 1
9 0 0 0 1
10 0 0 0 1
11 0 0 0 1
12 + 0 0 0 1

```

MM->2->4

```

Spl- 1<>P..1...2...3...4...5...6...7...8
Time 15 50 15 20 15 50 15 20
Coor-P . X . . . . .
Mode NON MAX NON NON NON MAX NON NON

```

MM->2->7->1

```

Ely Off Ret Min >
Pt Sht Lng Dwl Yld Ref Hld Flt V P % MI
1 10 25 0 0 End X . . . . .
2 0 17 0 0 Beg . . . . .
3 0 17 0 0 Beg . . . . .
4 0 17 0 0 Beg . . . . .
5 0 17 0 0 Beg . . . . .
6 0 17 0 0 Beg . . . . .
7 0 17 0 0 Beg . . . . .
8 0 17 0 0 Beg . . . . .
9 0 17 0 0 Beg . . . . .
10 0 17 0 0 Beg . . . . .
11 + 0 17 0 0 Beg . . . . .

```

MM->2->5

```

< No Shortway 1 2 3
Pt 12345678 90123456 78901234 56789012
1 .....
2 .....
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....
11 + .....

```

MM->2->5

Running this in the controller with constant calls on each phase and peds 2,4,6, and 8, the following is the calculated values shown under **MM->2->8->2**, Easy calcs.

Easy <>	P..1...2...3...4...5...6...7...8
PrimFrc	51 0 16 35 51 0 16 35
SecdFrc	51 0 16 35 51 0 16 35
Veh Yld	0 10 0 0 0 10 0 0
VehAply	42 76 7 21 42 76 7 21
Ped Yld	0 10 0 0 0 10 0 0
PedAply	46 95 11 30 46 95 11 30
FloatMx	10 44 10 14 10 44 10 14
PedLeav	51 90 16 25 51 90 16 25
PedCall	46 95 11 30 46 95 11 30
SpltRem	0 0 0 0 0 0 0 0

The coordinator will stop when running ped phases 4 & 8 and force a short way transition within the cycle. Keep in mind that the Stop-In-Walk will work on both side road and artery phases. To control recycles of the peds, especially arterial pedestrians, the user can follow one or both of the strategies below.

### **Min P**

Use Min P selection under MM->2->5. Setting the Min P selection will not change the Easy Calcs screen but will hold off recycling of the artery peds and avoid a late ped call coming into the side road.

### **Walk Recycle**

As shown in the Ped Recycle example setting the Artery Pedestrians to Rest-in-Walk and using the Walk recycle settings will also assist in the control of the pedestrians.

The user should be reminded that Stop-in-walk should only be used when pedestrian actuations are infrequent. Stop-In-Walk is enhanced by short-way offset correction because the coordinator can usually re-synchronize the offset within one cycle when ped clearance only extends 5 – 10” beyond the force-off.



# 2 OTHER Coordination Modes

Chapter 6, Basic Coordination in the Cubic | Trafficware controller manual shows how to program *Split Tables* for NTCIP FIXED and FLOATing force-offs. This section explains how to program *Split Tables* for the seven OTHER coordination modes. These OTHER methods program individual force-off and yield points that are automatically calculated using FIXED and FLOAT and provide additional control over the permissive windows. The disadvantage of the OTHER modes are that they are more complex to program and do not provide many of the coordination diagnostics discussed in Chapter 6 for FIXED and FLOAT. *Easy Mode* is an exception and because it is so similar to the FIXED and FLOAT NTCIP methods.

## 2.1 Easy

Spl-24	Ø	1	2	3	4	5	6	7	8	->
Time	20	20	20	20	20	20	20	20	20	
Coor-Ø	.	X	.	.	.	.	.	.	.	
Mode	NON	MAX	NON	NON	NON	MAX	NON	NON		

### MM->2->7->1: Split Table - 80" Cycle (STD8) Programmed for FIXED or FLOAT

The *Easy Coordination Mode* has two variations depending if *Easy Float* under *Coordination Modes+* (MM->2->1, right menu) is set ON or OFF. This mode with *Easy Float* OFF is very similar to the NTCIP FIXED force-off method discussed in the last section. *Easy Mode* with *Easy Float* ON is very similar to the NTCIP FLOAT method.

The differences between the NTCIP modes and the *Easy Mode* of coordination are as follows:

- The offset is always referenced to *Begin-of-Green* of the *Coordinated Phase* (the NTCIP offset reference under MM->2->5, right menu, does not apply in *Easy Mode*)
- Yield points are more constrained. That is, the “windows of opportunity” to service the non-coordinated phases are opened later in the cycle than the NTCIP methods which yield to the non-coordinated phases when the coordinated phase is forced off

The following *Easy Calcs* are generated for the 80" cycle shown in the above split table. Compare these values with the *Easy Calcs* for the FIXED and FLOAT example in section 6.3.2 for *Begin-of-Green*. Note, that the force-off points are set one second earlier in *Easy*. Also note that the *Yield Points* for the non-coordinated phases in FIXED and FLOAT are all set at the end of the coord phase, but are staggered in *Easy*. The *Yield Points* in *Easy* allow the coordinator to dwell in the coord phase if the next phase is skipped. This provides “slack time” to the end of the progression phases when the next phases are skipped.

Easy	Ø	1	2	3	4	5	6	7	8 ->
PrimFrc	74	14	34	54	74	14	34	54	
SecdFrc	74	14	34	54	74	14	34	54	
Veh Yld	22	26	14	18	22	26	14	18	
VehAply	65	5	25	45	65	5	25	45	
Ped Yld	22	26	14	18	22	26	14	18	
PedAply	74	5	34	45	74	5	34	45	
FloatMx	15	15	15	15	15	15	15	15	
PedLeav	74	4	34	44	74	4	34	44	

MM->2->8->2: Easy Calcs for the 80" Cycle Shown Above – Easy Mode (Begin-of-Green)

## 2.2 Perm,Frc

The Perm,Frc parameters that will allow the user to override the calculated settings for coordination. If a user decides to use this to modify the coordination calculations, he/she should select all settings and change them using the existing Easy Calcs as a base for this change. In addition the user must test the changes to insure the desired operation.

### 2.2.1 Primary Force-Off

The *Primary Force-Off* is the point in the local cycle that a force-off is applied to a phase causing that phase to terminate and begin timing yellow clearance. A *Primary Force-off* will remain applied until the phase terminates. It is up to the user to insure that *Primary Force-Offs* are applied after the minimum phase times of each phase.

The coordination diagnostics does not check minimum phase when force-offs are programmed directly like the FIXED and FLOAT coordination methods. **Therefore, it is possible to program force-offs incorrectly and skip phases.** Care must be taken to ensure that each force-off needs to accommodate the split times including any pedestrians that are programmed. If the phase is skipped for three cycles in a row, the coordinator will fail the pattern. Coord diagnostics provided with FIXED and FLOAT detect these errors before the pattern is run and place the controller in a FREE fail condition.

Sp1- 1 En<	1	2	3	4	5	6	7	8>
HoldToMax	.	.	.	.	.	.	.	.
PriFrc	0	0	0	0	0	0	0	0
VAply	0	0	0	0	0	0	0	0
VehYld	0	0	0	0	0	0	0	0
PAply	0	0	0	0	0	0	0	0
PedYld	0	0	0	0	0	0	0	0

	Beg	End	7890123456789012
Perm1	0	0	.....
Perm2	0	0	.....
Perm3	0	0	.....
FrcAll	0		
PedRcy	0		

### 2.2.2 HoldToMax (V80.x or V85.x/Scout only)

This parameter will force a phase to be held to its programmed maximum time or the calculated force off if it has a call. This is used for side road phases to ensure that when there is a call, they stay on and are not subject to gapping out.

### 2.2.3 VApply

This parameter allows the user to modify the vehicle apply point for each phase. Please take under consideration when modifying each phases apply point.

The Vehicle Apply point (VehAply) for each phase is typically calculated as:

$$\text{Vehicle Apply Point (VehAply)} = \text{Primary Force-off} - ((\text{Max Yellow} + \text{All Red}) + \text{Minimum Green})$$

The yield point must be earlier than the automatic application point for the phase to be serviced. If short-cycle offset correction is enabled, the yield point must be earlier still to allow for the effective reduction in split time that occurs when the local cycle timer corrects by running fast.

### 2.2.4 VehYld

The *Vehicle Yield* is that point in the cycle that a vehicle call on a phase will be serviced, i.e. that the phase's inhibit is removed. Note that the phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off. The Vehicle Apply point (VehAply) for each phase is calculated as:

$$\text{Vehicle Apply Point (VehAply)} = \text{Primary Force-off} - ((\text{Max Yellow} + \text{All Red}) + \text{Minimum Green})$$

The yield point must be earlier than the automatic application point for the phase to be serviced. If short-cycle offset correction is enabled, the yield point must be earlier still to allow for the effective reduction in split time that occurs when the local cycle timer corrects by running fast.

### 2.2.5 PApply

This parameter allows the user to modify the pedestrian apply point for each phase. Please take under consideration when modifying each phases apply point.

The PedApply point for each pedestrian phase is calculated using the following table:

Ped Apply Calculation ( all calculation parameters are in seconds)	Modified Parameter (s)
PedApply = PrimaryForceOff - [ PedClearance + (Walk - 5) (if Walk >5) ] + 1 ( if PedClearance >0 )	Stop-in Walk=OFF
Note:	Rest-in Walk = OFF
One possible reason for Walk -5 would be to make the end of Ped Clearance aligned with the end of Yellow or All-Red because Yellow+All-Red is usually 5 sec	Min P (Min Ped Permissive) = OFF
PedApply = PrimaryForceOff - 5	Stop-in Walk=ON
PedApply = PrimaryForceOff - ( Walk + PedClearance ) + 1 ( if PedClearance >0 )	Rest-in Walk = ON
PedApply = PrimaryForceOff + 5 - ( Split Time - Yellow - All-Red ) + 1 ( if PedClearance >0 )	Min P (Min Ped Permissive) = ON
PedApply = PedApply - 1 - ( Yellow + All-Red )	Stop-in Walk=OFF and WALK2 is programmed

The same considerations described above for selecting vehicle yield points apply to determining pedestrian yield points except when the STOP-IN-WALK is enabled. Refer to the explanation of Stop-In-Walk.

### 2.2.6 Pedestrian Yield

The *Pedestrian Yield* is that point in the cycle that a pedestrian call on a phase will be serviced, i.e. that the phases pedestrian inhibit is removed. The phase inhibit is automatically applied by the controller at a calculated time in advance of the primary force-off per the *PedApply* point shown is the above table.

The same considerations described above for selecting vehicle yield points apply to determining pedestrian yield points except when the STOP-IN-WALK is enabled. Refer to the explanation of Stop-In-Walk.

### 2.2.7 Permissives

The Permissive method allows you to specify up to three permissive “windows of opportunity” to service the yield phases programmed in the *Split Plus Features*. Programming these periods where you allow phases these windows can assist the user in complicated intersections.

### 2.2.8 FrcAll

This is an entry which allows selection of a point along the coordinated cycle that will cause a force-off on any phase which is green. This is programmed in seconds from 0-255.

### 2.2.9 PedRcy

This entry activated when timing the permissive mode in seconds as the point along the coordinated cycle when the coordinated phase(s) recycles to walk

	Beg	End	7890123456789012
Perm1	0	10	.....
Perm2	0	0	.....
Perm3	0	0	.....
FrcAll	40		
PedRcy	30		

# 3 External Input / Output Programming

*External I/O* allows an external source to select the active pattern using *Offset* and *Plan* inputs provided on the D-connector. External coordination schemes date back to early TS1 days when an on-street master selected the active pattern of all secondary controllers in the system through an AC current based hardware interconnect

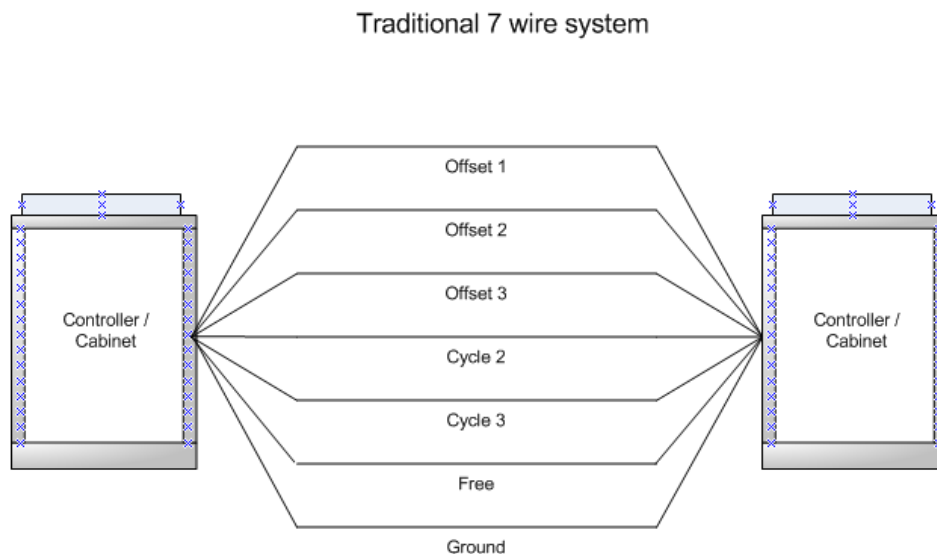
*External I/O* programming is provided for backward compatibility

with these older systems. The *External I/O* programming shown to the right associates the *Offset / Plan* inputs with the NTCIP pattern provided in the pattern table.

Pat#	Offset	Plan	Pat#	Offset	Plan
1	1	1	2	1	1
3	1	1	4	1	1
5	1	1	6	1	1
7	1	1	8	1	1

## 3.1 Setting up External I/O

Legacy hardwire systems utilize channel outputs from one controller that are hard-wired to isolated inputs of another controller as shown below.



Typical equipment and coordination system software will run 3 cycles and 3 offsets. For a system that is using a 7 wire scenario, there are wires for offsets 1, offset 2, offset 3 cycle 2 and cycle 3. These wires are connected from a master cabinet's output rack to a slave cabinet's input rack. If no cycle input is on, then the software assumes that the default cycle, cycle 1, is active. The synchronization (sync) pulse is generated when the offset input is turned off for a short period of time (typically 1 second). There is also a ground wire and a wire to force free actuated control. If no sync pulse is received for 3 cycles in a row, the controller will revert to free operation.

These wires can be utilized in an NTCIP controller that is connected to a legacy master. However, NEMA and NTCIP specifications do not use cycle inputs or outputs. For a local controller receiving information from a legacy master, it will use 4 inputs; Timing Plan A, Timing Plan B, Timing Plan C and Timing Plan D. These inputs, together with the Offset inputs will select a NTCIP pattern number that will control coordination. It is up to the user to properly map these inputs to get their desired patterns that the legacy system wants implemented.

## 3.2 External Plan Setup

The NEMA TS2 specification, which Cubic | Trafficware adheres to have set up the four Plan inputs (Plan A, Plan B, Plan C and Plan D) to choose a NEMA Plan number (0-15) as shown in the table below. Cubic | Trafficware uses a slightly different set of Plan numbers for simplification of External I/O programming:

Original Naztec Plan #	NEMA Plan #	Plan B Input	Plan A Input	Plan D Input	Plan C Input
1	0	Off	Off	Off	Off
2	1	Off	Off	Off	ON
3	2	Off	Off	ON	Off
4	3	Off	Off	ON	ON
5	4	Off	ON	Off	Off
6	5	Off	ON	Off	ON
7	6	Off	ON	ON	Off
0	7	Off	ON	ON	ON
9	8	ON	Off	Off	Off
10	9	ON	Off	Off	ON
11	10	ON	Off	ON	Off
12	11	ON	Off	ON	ON
13	12	ON	ON	Off	Off
14	13	ON	ON	Off	ON
15	14	ON	ON	ON	Off
16	15	ON	ON	ON	ON

## 3.3 External I/O Programming Steps

### 3.3.1 General Information

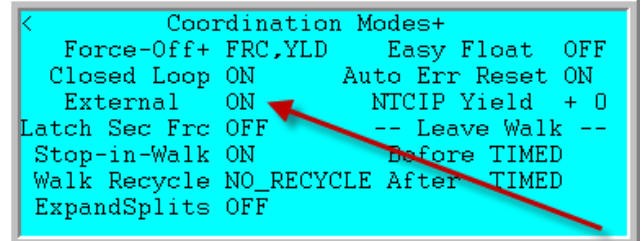
External I/O will only work when the scheduler is turned off. The reason for this is that an external hardware source is controlling local controller. Therefore do not program the scheduler.

When using a version 980 controller, the user must depend on NEMA mapping to map the External I/O. Many times the user will use a particular D-Connector Map.

In a 2070 Type controller, the user can remap Inputs or outputs or use Logic I/O to map External I/O.

### 3.3.2 Coordination Menu (MM->2)

Under coordination, first set up the Modes (MM->2->1) to accept External IO inputs.



Next set up The External I/O Table (MM->2->2) to choose up to 32 patterns based on the Timing Plan Inputs and Offset input combination, that you, the user, define.

Pat#	Offset	Plan	Pat#	Offset	Plan
1	1	1	2	1	2
3	1	3	4	1	4
5	1	5	6	1	6
7	1	7	8	1	8
9	1	9	10	1	10
11	1	11	12	1	12
13	+	1	13	1	14

Pat#	Offset	Plan	Pat#	Offset	Plan
15	-	1	16	1	16
17	2	1	18	2	2
19	2	3	20	2	4
21	2	5	22	2	6
23	2	7	24	2	8
25	2	9	26	2	10
27	+	2	28	2	12

29	2	13	30	2	14
31	2	15	32	2	16

The Plan column corresponds to the “Naztec” Plan Number. Which is selected based on Plan Input A, Plan Input B, Plan Input C or Plan Input D as shown in the table described in section 3.2. The Offset column corresponds to the Offset input 1, Offset Input 2 or Offset Input 3. The coordination pattern will be selected based on this combination.

### 3.3.3 Plan Selection Examples

- 1) If the controller has all Plan Inputs off but has Offset input 1 On, Pattern 1 will be chosen.
- 2) If the controller sees Plan C input ON and Offset 2 Input On, and all other inputs OFF, Pattern 18 will be selected.
- 3) If the controller sees Plan A, Plan B, Plan C and Plan D inputs On and Offset 1 Input On, Pattern 15 will be selected.