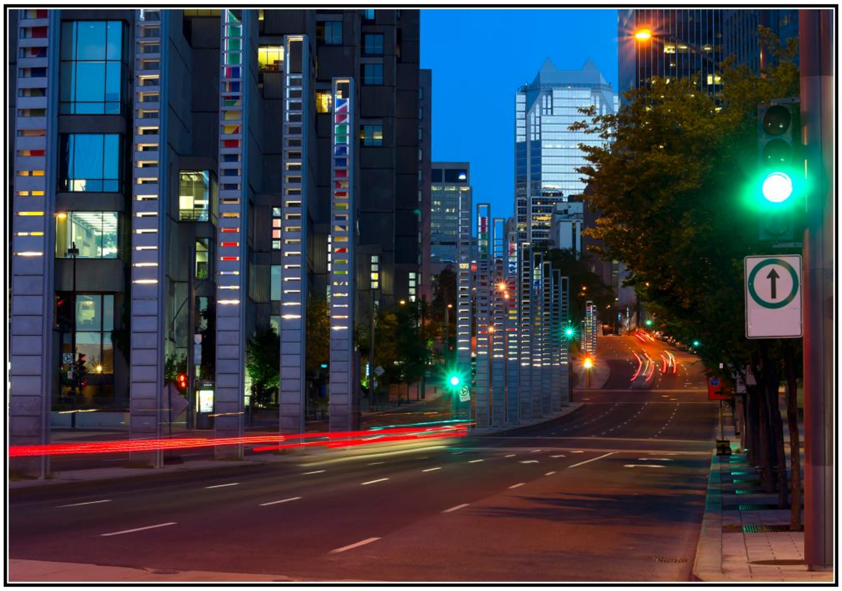
Operations Manual

For

Advanced Coordination Topics for Controllers Using V61.x or V65.x



February 2021

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

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:

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

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 Cubic | Trafficware representative for further details.

1 Coordination Programming Considerations and Best Practices

NTCIP Coordination at its best allows users to simply program coordination for a corridor of intersections. As Chapter 6 of the V61.x, V65.x, V76.x, V80.x or V85.x/Scout NTCIP controller manuals describe, 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.

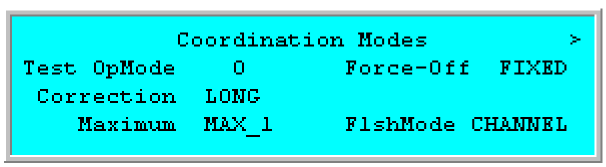
Note: When considering coordination, using the STD8 phase mode will take advantage of the most coordination diagnostic checks to catch common data entry mistakes, and if detected, times the intersection in FREE. In USER mode, most of these coordination diagnostics are removed, and the onus is on the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

1.1 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 V61.x, V65.x, V76.x V80.x or V85.x/Scout NTCIP controller manuals.

1.1.1 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**).

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.

1.1.2 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.

FLOAT 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.

FIXED 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 **Maximum Mode** for force off point calculations.

OTHER 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.

1.1.3 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.

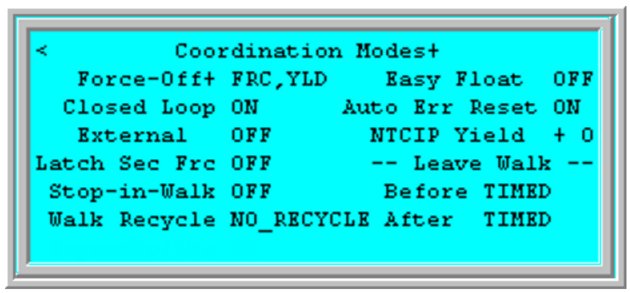
MAX_1 Selecting the MAX_1 mode allows *Maximum 1* phase timing to terminate a phase when FIXED or OTHER force-off methods are in effect.

MAX_2 Selecting the MAX_2 mode allows *Maximum 2* phase timing to terminate a phase when FIXED or OTHER force-off methods are in effect.

MAX_INH Selecting MAX_INH inhibits *Maximum 1* and *Maximum 2* 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.

1.1.4 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.

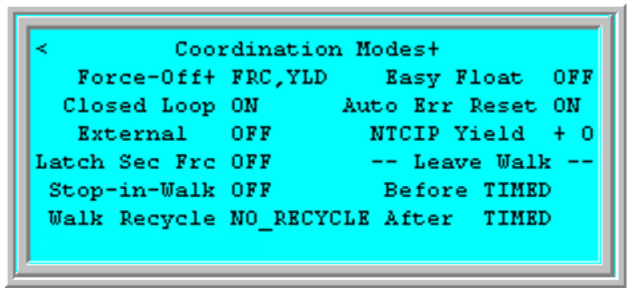
OFF *Stop-in-Walk* OFF forces the user to provide adequate split time to service the walk and ped clearance intervals assigned to the phase. The coordination diagnostic will fail the pattern if the split times do not adequately meet the pedestrian requirements.

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.

1.1.5 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 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.

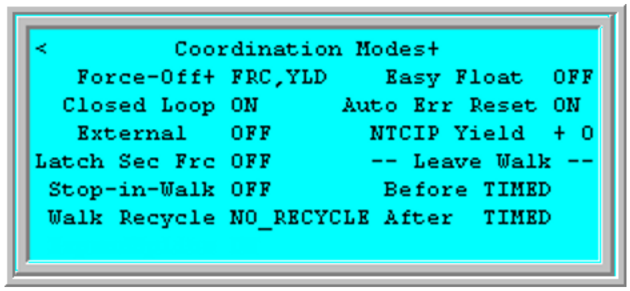
Ø1256_INH This option is useful when the coord phase is Ø4 or Ø8. The coord phase walk is not recycled until the permissive window for the cross street (Ø1256) has had an opportunity to service conflicting pedestrian and vehicle calls.

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 reservicing 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.

1.1.6 Leave Walk Before / Leave Walk After



These parameters apply to the artery phases resting in walk before being the pedestrian is 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:

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.

1.1.7 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.

1.1.8 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.

1.1.9 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.

1.1.10 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.

1.1.11 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.

1.2 Easy Calcs Status Screen (MM->2->8->2)

When programming NTCIP coordination, the user has the ability to see what parameters have been calculated by the coordination algoritmin. 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
SpltRem	0	0	0	0	0	0	0	0

1.2.1 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.

1.2.2 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.**

1.2.3 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.

1.2.4 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.

1.2.5 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.

1.2.6 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.

The PedApply point for each pedestrian phase is calculated as:

Ped Apply Point (PedAply) = Primary Force-off – ((Max Yellow + All Red) + Pedestrian Clear)

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.

1.2.7 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.

1.2.8 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.

1.2.9 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)

1.2.10 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.**

1.3 Transition Considerations using NTCIP settings

1.3.1 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 **BeginGrn** / **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 S**TOP-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 then 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.

1.3.2 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.

1.3.3 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 t**ransition 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 spit 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 S**hortway** 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. **Dynamic Shortway** waits for the controller to be in transition. The software then does the following:

a) It looks at all the phases that are ON

b) For all phases ON, it calculates the largest **Shortway percentage** that phase can run and not it to violate its minimums.

c) It chooses the ***SMALLEST*** **Shortway** percentage that is calculated for each phase "ON" (because otherwise a larger one would violate the smaller one)

d) Once **Dynamic Shortway** is set to ON, a **Shortway** percentage must be programmed in each pattern. Setting the **Shortway** percentage to 1% will allow the algorithm to process.

e) Since this is **Dynamic Shortway** keep in mind that your ability to transition is controlled by which phases are running. Therefore, if a phase 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 thru, and the left turn does not have a lot of slop time, then the thru will be constrained until the left turn terminates.

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.

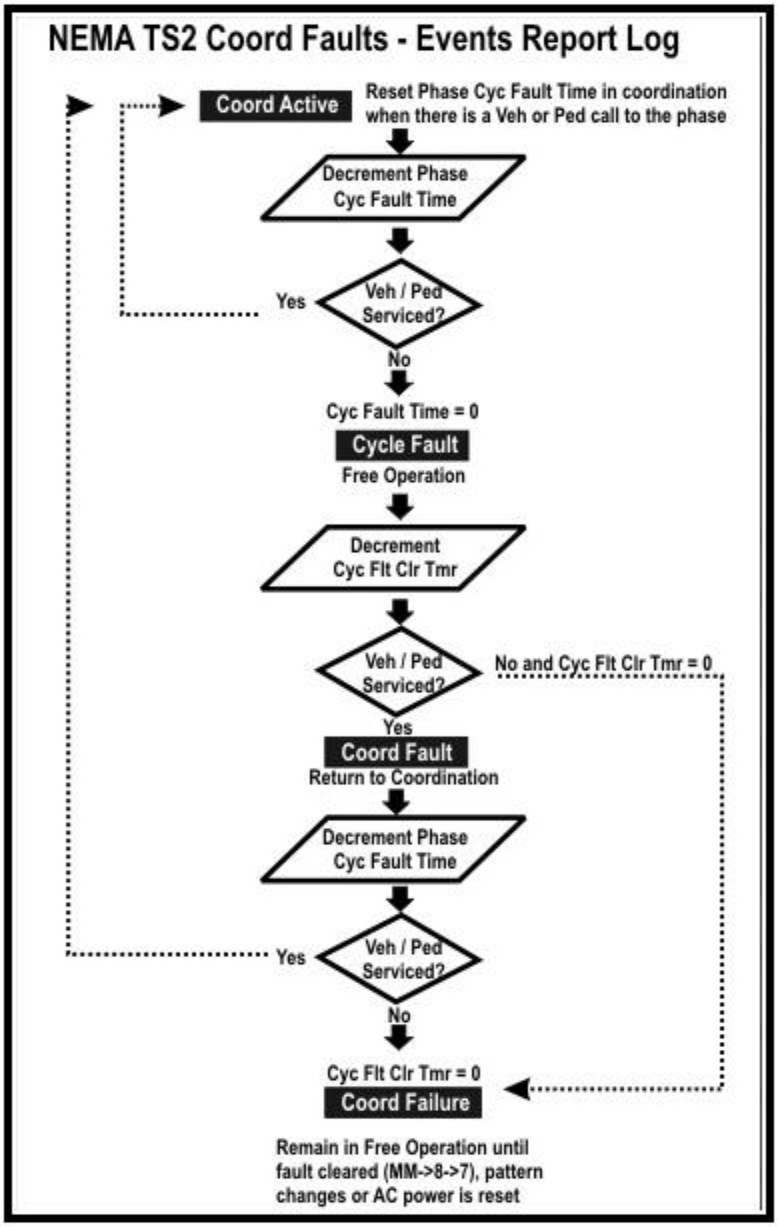
1.4 Coordination Alarm Considerations

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.
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 (V76.x, V80.x, V85.x/Scout only)
61	Coordination in (Sync) Transition	(V76.x and V80.x, V85.x/Scout only) 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.

1.4.1 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:

a) In all cases a "***cycle fault***" is declared.
b) If the controller is running free then a "***cycle failure***" occurs
c) If the controller is running coordination then a "***coord cycle fault***" will occurs on the first occurrence of a cycle fault.
d) 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.
e) 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 (a phase is not service for the fault timer amount of time) for any reason, the Cycle Fault is 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 Fail 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.

1.4.2 Alarm 30 Pattern Error Faults

Fault #	Fault Description
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

1.5 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.

1. The sum of splits must equal cycle length in all rings.
2. Within a barrier, splits in ring 1 must equal ring 2 and equal ring 3, etc.
3. There must be a phase in each barrier in each ring that is used.
4. The reference point must touch a barrier.
5. User testing. The user must:
    a. Run all patterns to verify phase checks and easy calcs
    b. Run all patterns with max recall to verify correct splits
    c. 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.
6. The user is ultimately responsible for checking the Easy Calcs and verifying the correct operation at the intersection.

1.6 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 coord 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 coord reference phase, you simply get a return hold.

- Walk Recycle selections (specifically the "1256/3478 inhibit) mode works the same.

- Coord 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.

1.7 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.

1.7.1 Basic program

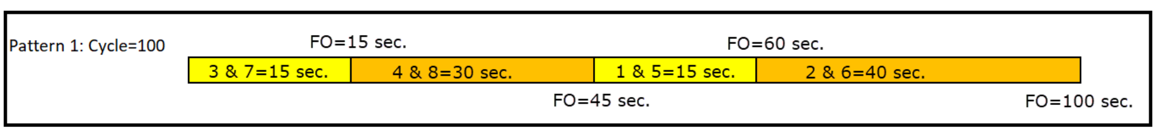
The intersection that will be modified is a standard 8 phase intersection running sequence 1 as shown below:

Ring 1	1	2	3	4
Ring 2	5	6	7	8

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 and V85.x/Scout are shown below. V61.x and V65.x features are programmed the same.

```
< Options        ..1..2..3..4..5..6..7..8>
        Enable P #  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+.  .  .  .  .  .  .  .

      MM->1->1->2
```

```
Times < >..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

      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 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

      MM->2->1
```

```
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  40  15  30  15  40  15  30
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	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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

1.7.2 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
       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:

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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

Modified 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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

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

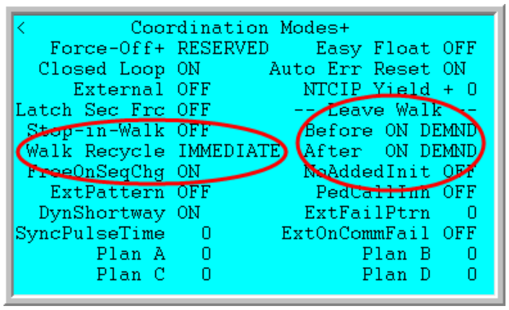
Original calcs

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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

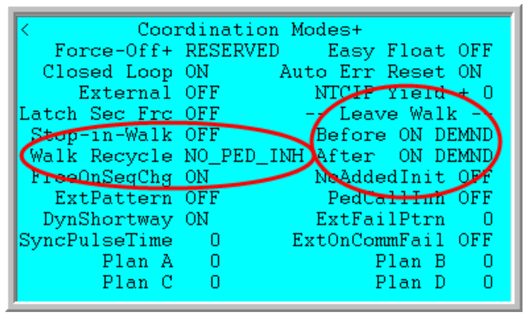
Modified Calcs

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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

Now change the Leave walk parameters to ON DEMND and select Walk Recycle as IMMEDIATE or NO_PED_INH.



or



The easy calcs are calculated as shown below.

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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

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
VehAply	52	76	7	31	52	76	7	31
Ped Yld	0	10	0	0	0	10	0	0
PedAply	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

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.

.

1.7.3 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, 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>
Enable P	#	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+	.	.	.	.	.	.	.	.

MM->1->1->2

Times < >	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								

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 Frc 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
```

```
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 CIC (Critical Intersection Control)

The *Advanced Coordination* topics extend the *Basic Coordination* NTCIP methods discussed in Chapter 6. These advanced methods are purely optional and add more complexity to the programming required to define a coordination pattern. The *Basic Coordination* methods described in Chapter 6 are adequate for most situations and combine ease of use with extensive diagnostics to insure reliable operation. However, in some instances, *Advanced Coordination* methods may be desirable to provide:

- greater control over the management of "slack time" in the controller
- greater control over the permissive windows of opportunity for the non-coordinated phases
- the ability to force-off the same phase twice per cycle

CIC (Critical Intersection Control) is easy to program and allows "slack time" to be managed by a dynamic split adjustment performed each cycle. The OTHER modes of *Advanced Coordination* discussed in this chapter require the user to manually program force-offs, permissive yield and apply points. The *Basic Coordination* methods defined in Chapter 6 automatically calculate these *Easy Calcs* for you when the split times and sequence are specified for the pattern.

CIC (Critical Intersection Control) is an optional *adaptive split* feature used with NTCIP FIXED force-offs. This feature is enabled by programming one of four *CIC Plans* under the coordination CIC menu (either MM->2->3 or MM->2->9->2 (E-Now) and associating the *CIC Plan* with a pattern in MM->2->6 as shown below.

CIC#	CoorØ	Grow	Ø.1	..2	..3	..4	..5	..6	..7	..8
1	OFF	0	0	0	0	0	0	0	0	0
2	P26	1	0	6	3	3	0	6	3	3
3	P48	2	3	6	3	3	3	6	3	3
4	P2468	1	3	6	3	6	3	6	3	6

MM->2->3 : CIC Plans (Numbered # 1 - #4)

CIC#	CorØ	Gr/ASC	Ø.1	..2	..3	..4	..5	..6	..7	..8
1	OFF	0	0	0	0	0	0	0	0	0
2	P26	1	0	6	3	3	0	6	3	3
3	P48	2	3	6	3	3	3	6	3	3
4	P2468	1	3	6	3	6	3	6	3	6

MM->2->9-2: CIC Plans (Numbered # 1 - #4)

<-	Pat#	Olp.Off:12345678	CIC	CNA1	Max2	Dia
	1	....	0	.	.	DFT
	2	....	2	.	.	DFT
	3	....	3	.	.	DFT
......						
	48	....	4	.	.	DFT

MM->2->6 (right menu): An Optional CIC# is Associated with Each Pattern #

CIC modifies split times by adjusting force-offs once per cycle in coordination using a method similar to the *Dynamic Max* in free operation. The *CIC* algorithm reduces *Split Times* for phases that gap-out after two consecutive cycles and distributes this time to the *Coord Ø(s)* or other “Grow” phases listed in the *CIC* table. CIC insures that “Slack time” from the non-coordinated phases moves to the end of the coord phase rather than the next phase in the sequence. CIC improves progression by moving “slack time” to the end of the Coord-Ø instead of at the beginning of the Coord-Ø.

2.1 Example Using CIC (Critical Intersection Control)

This section provides a step-by-step example of CIC in operation and explains how to interpret the *CIC Calcs* status display and observe the dynamic split adjustments cycle by cycle.

Step 1 – Initialize the controller and modify the STD8 defaults

a) Turn the *Run Timer* OFF (MM->1->7) and initialize the controller as STD-8Ø under MM->8->4->1. Don't forget to turn the *Run Timer* back ON (MM->1->7)

b) Change the *Min Green* times phases 1 - 8 to 2" under MM->1->1->1. Also, change the *Yellow* time of each phase to 2" and the *All-Red* clearance times to 0". These changes will allow you to observe CIC operation quickly using a 40" cycle.

c) Set STOP-IN-WALK to ON under MM->2->1 (right menu). This setting allows *Split Times* to run shorter than the pedestrian minimums set for the through phase defaults.

Step 2 – Create the timing patterns

a) Create three timing patterns in the Pattern Table (MM->2->4) as shown below.

Pat#	Cycle	Offset	Split	Seqnc
1	40	0	1	1
2	40	0	2	1
3	40	0	3	1

MM->2->4: Timing Patterns Used for CIC Examples

b) Create the three *Split Tables* shown below. Pattern 1 is an example of coordination provided along one street phased on 2 and 6 (notice the Coord-Ø setting and the MAX recall applied to 4 and 8). Pattern 2 provides coordination to phases 4 and 8. Pattern 3 provides coordination to both intersecting streets (2, 4, 6 and 8).

Spl-1	Ø..1	...2	...3	...4	...5	...6	...7	...8 ->
Time	10	10	10	10	10	10	10	10
Coor-Ø	.	X	.	.	.	.	.	.
Mode	NON	MAX	NON	NON	NON	MAX	NON	NON

Pattern 1 / Split Table 1: Major arterial is situated on phases 2 and 6

Spl-2	Ø..1	...2	...3	...4	...5	...6	...7	...8 ->
Time	10	10	10	10	10	10	10	10
Coor-Ø	.	.	.	X	.	.	.	.
Mode	NON	NON	NON	MAX	NON	NON	NON	MAX

Pattern 2 / Split Table 2: Major arterial is situated on phases 4 and 8

Spl-3	Ø..1	...2	...3	...4	...5	...6	...7	...8 ->
Time	10	10	10	10	10	10	10	10
Coor-Ø	.	.	.	.	.	X	.	.
Mode	NON	MAX	NON	MAX	NON	MAX	NON	MAX

Pattern 3 / Split Table 3: Two major arterials crossing on phases 2, 4, 6 and 8

Step 3 – Assign “slack time” from the actuated phases

In these examples, a single Coord-Ø is used in the split table to reference the pattern offset to the beginning of the Coord-Ø green. This is the standard default; however, the offset reference may be changed to *EndGrn* under MM->2->5 (right menu). In these examples, the offset (zero point in the cycle, or Loc = 0) is synched to the beginning of the Coord-Ø specified in the *Split Table.*

The Coord-Ø is typically “fixed” this is the portion of the cycle that needs to be guaranteed for the progression (green bands) along the major street. *Return Hold* may be set under MM->2->5 (right menu) to insure that when the controller returns to the Coord-Ø that it holds the phase until it is forced off. However, it is more convenient to simply place a MAX recall on the progression phases in the split table. Therefore, in these examples, the MAX mode setting indicates which phases are coordinated and the Coord-Ø is simply used to reference the offset to the beginning of one of these coordinated phases.

Non-coordinated phases are typically fully actuated, sot the Mode setting in the Split Table is typically set to NON (None) or MIN (Min recall). NTCIP coordination specifies that any unused (or “slack time”) from the non-coordinated phases is either passed to the next phase in the sequence (FIXED force-offs) or to the Coord- Ø (FLOAT-ing force-offs). These concepts were discussed in section 6.3.2. Please review this section before continuing with this example because CIC operation builds upon these two methods.

CIC calls for FIXED force-offs, so “slack time” is always passed to the next available phase in the sequence. However, CIC constantly monitors whether phases gap-out or max-out each cycle and dynamically adjusts the fixed force-offs each cycle to move “slack time” to phases that continue to max-out each cycle and the Coord-Ø

FLOAT-ing force-offs move all “slack time” from the actuated phases to the Coord-Ø. However, CIC is often preferred over FLOAT-ing force offs because “slack time” is move to the end rather than the beginning of the progression band. This reduces the early return problem common with FLOAT-ing force-off methods because “slack time” at the beginning of the progression band varies the start of the platoon.

There are essentially 4 ways to manage “slack time” in a semi-actuated controller during coordination:

1) **FIXED Force-offs Without CIC** – Force-offs are fixed and “slack time” is provided to the next phase.

2) **FIXED Force-offs With CIC** – Force-offs are adjusted dynamically to allocate “slack time” to phases that conitinue to max-out each cycle. Any remaining “slack time” allocated to the end of the Coord- Ø.

3) **FLOAT-ing Force-offs Without CIC** – A separate max timer insures that the non-actuated phases never time more than their programmed split. This moves “slack time” to the beginning of the Coord-Ø.

4) **OTHER Force-off Methods** – The OTHER methods are discussed in the last section of this chapter.

Step 4 – Associate a CIC Plan with each pattern

Any of the four *CIC Plans* may be assigned to the 48 patterns from *Alt Tables+* (MM->2->6, right menu). For these examples, associate *CIC Plan* 1 with pattern 1, *CIC Plan* 2 with pattern 2 and *CIC Plan* 3 with pattern 3 as shown below:

<-	Pat#	Olp.Off:12345678	CIC	CNA1	Max2	Dia
	1	....	1	.	.	DFT
	2	....	2	.	.	DFT
	3	....	3	.	.	DFT

MM->2->6: Coordination Alt Tables+

Step 5 – Program CIC Plans 1-3 called by Patterns 1-3

Program the first three *CIC Plans* associated with patterns 1-3 under MM->2->3 or 2->9->2 as shown below.

CIC#	CoorØ	Grow	Ø.1	..2	..3	..4	..5	..6	..7	..8
1	P26	2	0	20	10	10	0	20	10	10
2	P48	2	10	10	10	30	10	10	10	30
3	P2468	1	10	20	10	20	10	20	10	20
4	OFF	0	0	0	0	0	0	0	0	0

MM->2->3 or MM->2->9->2 : CIC Plans 1-3 called by Patterns 1-3

Note that the Coord-Ø specified in the *CIC Plan* table corresponds with the MAX mode settings set in the split tables. The MAX settings insure that split times for the coordinated (or progression) phases are guaranteed.

The “Grow” setting may range between 0 and 2 seconds (0 effectively defeats CIC). This parameter controls how much each split time is allowed to “grow” or “shrink” each cycle. The time specified under each phase is called the *Dynamic Max* because it controls the maximum adjustment (positive or negative) allowed for each phase. In the example above, the *Dynamic Max* for the coord phases in each ring is typically set to the sum of the *Dynamic Max* times for the non-coordinated phases. This allows all “slack time” from the non-coordinated phases to move to the end of the coord phases under CIC.

It is important to note that the *Dynamic Max* adjustment cannot reduce split times shorter than the minimum phase times. Note that split times in our example patterns are 10” while the *Dynamic Max* times are set 10-30”. CIC insures that split times are not reduced short enough to fail the pattern. Therefore, the *Dynamic Max* settings do not need to be checked by the coordination diagnostics. This simplifies the use of CIC and allows the same CIC Plan to be used for any number of patterns each with varying split times and cycle length.

Step 6 – Test Pattern 1

Force the controller to *Test Pattern 1* by setting *Test, OpMode* to “1” under MM->2->1. Observe the *CIC Calcs* screen under MM->2->8->3. When CIC begins, the *CIC Calcs* screen appears as follows:

Dyn Coor	Ø.1	...2	...3	...4	...5	...6	...7	...8
Dyn Acc	0	0	0	0	0	0	0	0
Dyn Abs	0	0	0	0	0	0	0	0
Dyn Max	0	20	10	10	0	20	10	10
DynTerm	0	0	0	0	0	0	0	0
PRIM FO	37	7	17	27	37	7	17	27
VEH YLD	7	17	7	7	7	17	7	7

MM->2->8->3: CIC Calcs When Pattern 1 Begins

When CIC begins, the primary force-off and vehicle yield points under *CIC Calcs* are identical to the FIXED force-offs under *Easy Calcs*. The zero point of the 40" cycle is at the beginning of the phase 2 (the Coord-Ø). Because the *Split Time* for phase 2 is 10" and yellow clearance is 3", phase 2 is forced-off at 10" – 3" = 7" after the offset. The controller yields to all non-coordinated phases when the coordinated phase is forced off.

Initially, the controller rests in phase 2 and 6 because MAX recalls are set for these phases in *Split Table 1* and there are no recalls present on the other phases. After 2 cycles, the *CIC Calcs* are recalculated and the "Grow" time of 2" is deducted from phases 3, 4, 7 and 8 and applied to phases 2 and 6 as shown below.

Dyn Coor	Ø.1	...2	...3	...4	...5	...6	...7	...8
Dyn Acc	0	4	254	254	0	4	254	254
Dyn Abs	0	4	2	0	0	4	2	0
Dyn Max	0	20	10	10	0	20	10	10
DynTerm	0	0	0	0	0	0	0	0
PRIM FO	37	11	19	27	37	11	19	27
VEH YLD	7	17	7	7	7	17	7	7

MM->2->8->3: CIC Calcs Two Cycles After Pattern 1 Begins

The *Dynamic Accumulator (Dyn Acc)* tracks the "grow" and "shrink" accumulations each cycle. Positive "slack time" is added to zero which serves as a base reference for positive accumulations. Negative "slack time" is subtracted from 256 as the base reference for negative accumulations. Therefore, a *Dyn Acc* equal to 254 is equivalent to a *Dyn Acc* value of –2 seconds.

After two cycles with MAX recalls on phases 2 and 6 and all other phases skipped, phases 3, 4, 7 and 8 "shrink" by 2" and phases 2 and 6 "grow" by 4". Notice how the 4" accumulation added to phases 2 and 6 extend the force-offs of phase 2 and 6 from Loc=7 to Loc=11 to extend the end of the coordinated movement.

After the third cycle (at Loc=0), another dynamic adjustment is made as shown below. Cross street phases 3, 4, 7 and 8 have "shrunk" a total of 4" each and the coordinated phases have grown by 8". The split times for phases 1 and 5 have not changed because the *Dynamic Max* values for these phases in the *CIC Plan* table are zero. You can easily control which *Split Times* are allowed to "grow" and "shrink" through the *CIC Plan* table

Note that at the end of cycle 3, the force-offs for phases 1 and 5 are still applied at 37; however, the force-off for phases 2 and 6 are at Loc=15 compared with cycle 1 (at Loc=7). These dynamic adjustments begin the coordinated phases at the same point in the cycle (at the end of phase 1 and 5), but extend the end of the coord phases using the "slack time" from the cross street phases.

Dyn Coor	Ø.1	2	3	4	5	6	7	8
Dyn Acc	0	8	252	252	0	8	252	252
Dyn Abs	0	8	4	0	0	8	4	0
Dyn Max	0	20	10	10	0	20	10	10
DynTerm	0	0	0	0	0	0	0	0
PRIM FO	37	15	21	27	37	15	21	27
VEH YLD	7	17	7	7	7	17	7	7

MM->2->8->3: CIC Calcs Three Cycles After Pattern 1 Begins

If you continue to observe this display, you will notice that no further split adjustments are made even though the controller continues to rest in 2 and 6 which all other phases are skipped. CIC cannot reduce split times shorter than the phase minimums plus a one second buffer. The sum of the minimum vehicle times is given by:

Minimum Vehicle Time = Min Green + Yellow + All-Red + 1" Buffer = 2" + 3" + 0" + 1" = 6"

A 10" split cannot "shrink" more than 4" without violating this *Minimum Phase Time* even though the *Dynamic Max* values in the *CIC Plan* table are 10". CIC guarantees the *Minimum Phase Times* are not violated, so the user need not be concerned about coord failures resulting from values in the *CIC Plan* table. Also, recall that STOP-IN-WALK was set ON in Step 1 c). If you turn STOP-IN-WALK OFF, all three patterns will fail the coord diagnostic to insure that the pedestrian min times are guaranteed:

Minimum Pedestrian Time = Walk + Ped Clear + Yellow + All-Red + 1" Buffer = 5" + 10" + 3 + 0" + 1" = 19"

Step 7 – Apply recalls to the non-coordinated phases in Pattern 1

Test Pattern 1 is currently resting in phase 2 and 6 with no calls on any of the non-coordinated phases. CIC has moved as much "slack time" as possible from phases 3, 4, 7 and 8 to the end of phases 2 and 6. In this step, we will observe how the "slack time" is transfers back to the non-coordinated phases as they are called into service.

Place MIN recalls on every phase but 2 and 6 using the Mode setting in *Split Table 1* as shown below. Confirm that these recalls are present from menu MM->7->1 and then watch the *CIC Calcs* from MM->2->8->3.

Spl-1	Ø..1	2	3	4	5	6	7	8 ->
Time	10	10	10	10	10	10	10	10
Coor-Ø	.	X	.	.	.	.	.	.
Mode	MIN	MAX	MIN	MIN	MIN	MAX	MIN	MIN

MM->2->7->1: Split Table 1 with MIN Recalls Applied to the Non-coordinated Phases

Note that after 4 or 5 cycles, the *CIC Calcs* have not changed from our last example even though all the non-coordinated phases are being recalled instead of being skipped. CIC cannot reduce the split times below the *Minimum Phase Times* as discussed in Step 6. Therefore, no split adjustments are made when the actuated service their min times.

Now go to *Split Table 1* and place a MAX recall on phase 4. Go back to the *CIC Calc* screen and observe how the Dynamic Max changes as phase 4 begins to "grow" back to it's original split time. These changes are summarized in the table below for 5 consecutive cycles.

Cycle #	Phase Recall Mode	CIC Dynamic Accumulator							
		1	2	3	4	5	6	7	8
1	4 MIN	0	8	252	252	0	8	252	252
3	4 MAX	0	6	252	254	0	6	252	254
4	4 MAX	0	4	252	0	0	4	252	0
5+	4 MAX	0	4	252	0	0	4	252	0

Example of Non-coordinated Phase 4 Regaining "Slack Time" and Growing Back to the Original Split

After 2 cycles with MAX recall applied to phase 4, the accumulated "slack time" provided to the end of the coordinated phases (phase 2 and 6) has been reduced and moved back to service phase 4. This dynamic split adjustment allows the *Split Times* to "grow" and "shrink" within the constraints of the *CIC Plan* table. This operation is similar to the *Dynamic Max* feature that allows max times to grow in a step-wise manner in free operation (see section 4.1.3).

Now, place a MAX recall on phase 3 in Split Table 1. You will observe the following dynamic split adjustment:

Cycle #	Phase Recall Mode	CIC Dynamic Accumulator							
		1	2	3	4	5	6	7	8
1	3 MIN and 4 MAX	0	4	252	0	0	4	252	0
3	3 & 4 MAX	0	2	254	0	0	2	254	0
4+	3 & 4 MAX	0	0	0	0	0	0	0	0

Example of Non-coordinated Phase 3 Regaining "Slack Time" Back to the Original Split

Step 8 – How CIC improves split utilization on the cross street

CIC can also make dynamic split adjustments to the non-coordinated phases to improve split utilization on the cross street. For example, assume the cross street left-turn movements are protected-only and that phases 3 and 8 max-out each cycle while phases 4 and 7 are running their *Minimum Phase Times*. We can simulate this condition by programming the *Mode* settings in *Split Table 1* as follows:

```
Spl-1   Ø..1...2...3...4...5...6...7...8 ->
Time      10  10  10  10  10  10  10  10
Coor-Ø     .   X   .   .   .   .   .   .
Mode     MIN MAX MAX MIN MIN MAX MIN MAX
```

Cross Street Phases 3 ad 8 are Max-out While Phases 4 and 7 Are Timing Their Mins

Observe the *CIC Calcs* until the dynamic splits adjustments come to rest as shown below.

Dyn Coor	Ø.1	..2	..3	..4	..5	..6	..7	..8
Dyn Acc	0	0	4	252	0	0	252	4
Dyn Abs	0	0	4	0	0	0	252	0
Dyn Max	0	20	10	10	0	20	10	10
DynTerm	0	0	0	0	0	0	0	0
PRIM FO	37	7	21	27	37	7	13	27
VEH YLD	7	17	7	7	7	17	7	7

CIC Improves Cross Street Split Utilization By Moving "Slack Time" Where It Is Needed

In this example, phases 1, 5, 2 and 6 continue to time their programmed *Split Times* (there is no dynamic split adjustment applied to the major street). However, phase 3 has gained an additional 4" from phase 4 and phase 8 has gained 4" from phase 7.

If FIXED force-offs without CIC was used in this example, phase 8 would receive the same unused "slack time" from phase 7 because phase 8 follows phase 7 in the sequence. However, phase 3 would never be allowed to "grow" by 4" using FIXED force-offs without CIC because phase 3 follows phase 2 in the sequence which is servicing it's MAXimum split. Therefore, CIC can improve split utilization for the first cross street phases serviced after leaving the coordinated street. This capability is especially useful if the cross street left-turns are protected only and max-out each cycle while "slack time" exists on the opposing through movements.

Step 9 – Test Pattern 2

Refer back to the *Split Table* for pattern 2 under Step 2. The Coord-Ø is phase 4 and MAX recalls are set on phases 4 and 8. This configuration would typically be used if the major street through movements are serviced on phases are 4 and 8 and the cross street is 2 and 6.

Now, force the controller into pattern 2 by setting *Test,OpMode* to "2" under MM->2->1.

Spl-2	Ø..1	...2	...3	...4	...5	...6	...7	...8	->
Time	10	10	10	10	10	10	10	10	
Coor-Ø	.	.	.	X	.	.	.	.	
Mode	NON	NON	NON	MAX	NON	NON	NON	MAX	

Pattern 2 / Split Table 2: Major arterial is situated on phases 4 and 8

This pattern calls for *CIC Plan # 2* as programmed under Step 5 and allows all non-coordinated phases to "grow" or "shrink" by 10". The coordinated phases can grow as much as 30" by applying "slack time" to the end of phases 4 and 8. Each phase is constrained by the *Minimum Phase Times* guaranteed for each phase.

CIC#	CoorØ	Grow	Ø.1	..2	..3	..4	..5	..6	..7	..8
1	P26	2	0	20	10	10	0	20	10	10
2	P48	2	10	10	10	30	10	10	10	30
3	P2468	1	10	20	10	20	10	20	10	20
4	OFF	0	0	0	0	0	0	0	0	0

MM->2->3: CIC Plans 1-3 called by Patterns 1-3

Test Pattern 2 rests in 4 and 8 because of the MAX recalls programmed for these phases in the *Split Table*. No other recalls are placed on any other phase in *Split Table 2*, so all the 12" of accumulated "slack time" is moved to the end of phases 4 and 8. This move the force-off points for phases 4 and 8 from Loc=7" to Loc=19".

Dyn Coor	Ø.1	...2	...3	...4	...5	...6	...7	...8
Dyn Acc	252	252	252	12	252	252	252	12
Dyn Abs	8	4	0	12	8	4	0	12
Dyn Max	10	10	10	30	10	10	10	30
DynTerm	0	0	0	0	0	0	0	0
PRIM FO	25	31	37	19	25	31	37	19
VEH YLD	7	7	7	17	7	7	7	17

In CIC Plan 1 we omitted the main street left-turns from the CIC split adjustment. *CIC Plan 2* allows the main street left-turn phases (3 and 7) to "grow" and "shrink" along with the cross street phases (3, 4, 7 and 8). If you place a MAX recall on phase 3 in *Split Table 2*, the *Dynamic Accumulator* changes as follows:

Dyn Coor	Ø.1	2	3	4	5	6	7	8
Dyn Acc	252	252	8	0	252	252	252	12

This example illustrates why you typically reduce or omit the main street left-turn phases from CIC. *Dynamic Max Time* for phases 3 and 7 should be constrained in *CIC Plan 2* to move more of the "slack time" to 4 and 8.

Step 10 – Test Plan 3 (Two Coordinated Intersecting Arterials)

This last example, both intersecting streets are coordinated. One of the phases is chosen as the Coord-Ø to reference the offset to the beginning of the Coord-Ø. MAX calls are placed on all through movements to guarantee cycle time to the progressed movements.

Spl-3	Ø..1	2	3	4	5	6	7	8 ->
Time	10	10	10	10	10	10	10	10
Coor-Ø	.	.	.	.	.	X	.	.
Mode	NON	MAX	NON	MAX	NON	MAX	NON	MAX

Pattern 3 / Split Table 3: Two major arterials crossing on phases 2, 4, 6 and 8

The Coord-Ø in CIC Plan # 3 is set to P2468 and the left-turn phases are set to "grow" or "shrink" a maximum of 10" compared to 20" for the progressed movements.

CIC#	CoorØ	Grow	Ø.1	2	3	4	5	6	7	8
1	P26	2	0	20	10	10	0	20	10	10
2	P48	2	10	10	10	30	10	10	10	30
3	P2468	1	10	20	10	20	10	20	10	20
4	OFF	0	0	0	0	0	0	0	0	0

MM->2->3: CIC Plans 1-3 called by Patterns 1-3

Force the controller into *Test Pattern 3* by setting *Test,OpMode* to "3" under MM->2->1. *Split Table 3* called by this pattern issues MAX recalls on phases 2, 4, 6 and 8 while phases 1, 3, 5 and 7 are skipped.

Observe the *CIC Calcs* under MM->2->8->3. Note that the *Dynamic Accumulator* adjustments are only 1" each cycle compared with the 2" adjustment in *Test Plans 1 and 2*. This is due to the difference in the *Grow* settings for these 3 CIC plans in the *CIC Plan* table.

3 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.

3.1 Easy

Spl-24 Ø	1	2	3	4	5	6	7	8	->
Time	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)

3.2 Permissive-Single (v61.x only)

Spl-24	Ø..1	...2	...3	...4	...5	...6	...7	..8 ->
Time	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

Split Times are entered in seconds and the *Coordinated Phase* and *Mode* settings are programmed like the FIXED and FLOAT methods discussed previously. The following Split+ Features are used to modify the force-off and yield points based on a single permissive window of opportunity for the non-coordinated phases.

This method provides a *Single Permissive* "window of opportunity" to service the non-coordinated phases. The beginning and end points for the permissive period are programmed through the *Split Plus* Features as shown below.

Spl- 1		Begin	End
Permissive		0	0
Frc_All	0	Ped-Recycle	0

MM->2->7->2: Split Plus Features, Single Permissive Period

3.3 Permissive Force-Off

Permissive Force-Off% (v61.x only)

Spl-24	Ø..1	...2	...3	...4	...5	...6	...7	...8 ->
Pri-Frc%	25	25	25	25	25	25	25	25
Coor-Ø	.	X	.	.	.	.	.	.
Mode	NON	MAX	NON	NON	NON	MAX	NON	NON
Sec-Frc%	25	25	25	25	25	25	25	25

MM->2->7->1: Split Table Using Permissive Force-Off % Coord Mode

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 do 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. 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.

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* defaults to the value of *Primary Force-off* whenever it is entered. However, the value of the force-off may be changed in the split table if needed.

The *Coordinated Phase* and *Mode* entries are the same as the FIXED and FLOAT modes defined in the last section. *Permissive Force-off%* mode is identical to the *Permissive force-off* mode, except primary and secondary force-offs are expressed as a percentage of cycle length (0-99%) instead of seconds.

The permissive force-off methods allow you to specify up to three permissive "windows of opportunity" to service the yield phases programmed in the *Split Plus Features* as shown below.

Spl- 1	Beg.	End	Yield.to.ø's							....
Perm-1	0	0	0	0	0	0	0	0	0	0
Perm-2	0	0	0	0	0	0	0	0	0	0
Perm-3	0	0	0	0	0	0	0	0	0	0
Frc_All 0	Ped-Recycle 0									

MM->2->7->2: Split Plus Features, Permissive Force-off Mode

3.4 Permissive-Float and Permissive-Float % (v61.x only)

Spl-24	ø..1	...2	...3	...4	...5	...6	...7	..8	->
% Time	25	25	25	25	25	25	25	25	
Coor-ø	.	X	.	.	.	.	.	.	
Mode	NON	MAX	NON	NON	NON	MAX	NON	NON	

MM->2->7->1: Split Table with Time in %

Permissive-Float% entries are the same as the NTCIP FLOAT mode, with the exception that time entries are expressed as a percentage of cycle length (0-99%) instead of seconds. These floating methods apply the *"FloatMx"* values to insure that "slack time" from the non-coordinated phases is passed back to the coordinated phase.

The *Permissive Float%* method allows you to specify up to three permissive "windows of opportunity" to service the yield phases programmed in the *Split Plus Features* as shown below.

Spl- 1	Beg.	End	Yield.to.ø's..........							
Perm-1	0	0	0	0	0	0	0	0	0	0
Perm-2	0	0	0	0	0	0	0	0	0	0
Perm-3	0	0	0	0	0	0	0	0	0	0
Frc_All 0	Ped-Recycle 0									

MM->2->7->2: Split Plus Features, Permissive Float%

3.5 Force-off / Yield (Frc/Yld) (v61.x only)

Spl-24	ø..1	...2	...3	...4	...5	...6	...7	...8	->
Pri-Frc	74	14	34	54	74	14	34	54	
Coor-ø	.	X	.	.	.	.	.	.	
Mode	NON	MAX	NON	NON	NON	MAX	NON	NON	
Sec-Frc	74	14	34	54	74	14	34	54	
Veh-Yld	22	26	14	18	22	26	14	18	
Ped-Yld	22	26	14	18	22	26	14	18	

MM->2->7->1: Split Table with Force-offs and Yields

The Force-off / Yield values in this menu are identical to the *Easy Calcs* generated for the 80" *Easy* mode pattern in section 6.4.1. The force-off, coord phase and mode settings of this menu are identical to the *Permissive Force-off* mode described in section 6.4.3. The additional yield settings provide full, independent control of the *Permissive Periods* and are described below.

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 do 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. 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.

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* defaults to the value of *Primary Force-off* whenever it is entered. However, the value of the force-off may be changed in the split table if needed.

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. The *Vehicle Apply* point (*VehApply* value under *Easy Calcs*) is calculated as:

Vehicle Apply Point (VehAply) = Primary Force-off – ((Max Yellow+Red of all Phases) + 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.

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 following calculation. This *PedApply* point is calculated as:

Ped Apply Point (PedAply) = Primary Force-off – ((Max Yellow + Red of all phases) + Pedestrian Clear + Walk)

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.

4 External Input / Output Programming

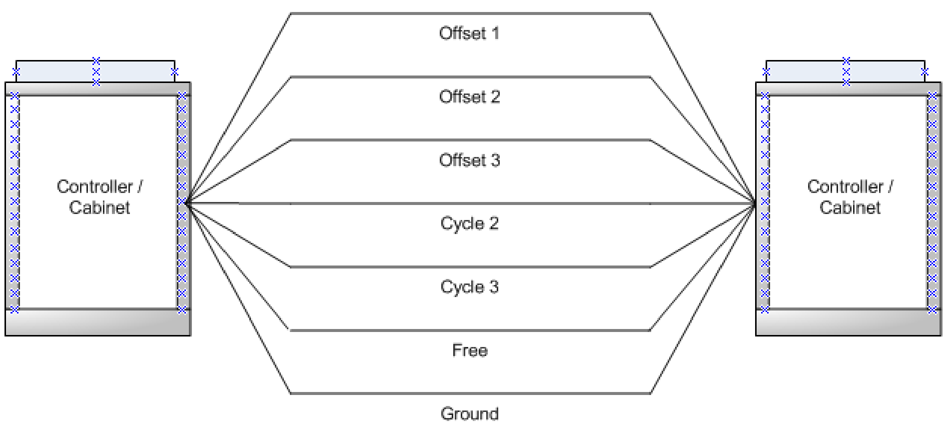
External I/O allows an external source to select the active pattern using *Offset* and *Plan* inputs provided on the D-connector. External coordination schemes date back to early TS1 days when an on-street master selected the active pattern of all secondary controllers in the system through an AC current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

4.1 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.

Traditional 7 wire system



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.

4.2 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

4.3 External I/O Programming Steps

4.3.1 General Information

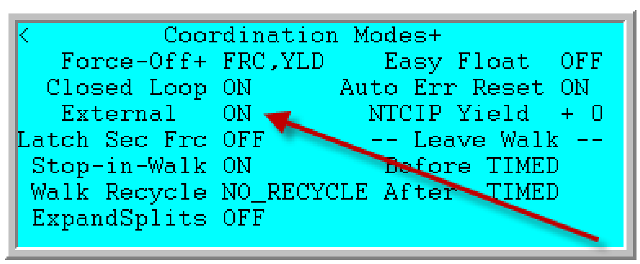
External I/O will only work when the scheduler is turned off. The reason for this is that an external hardware source is controlling local controller. Therefore do not program the scheduler.

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.

4.3.2 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	14	1	14
15 -	1	15	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	11	28	2	12
29	2	13	30	2	14
31	2	15	32	2	16

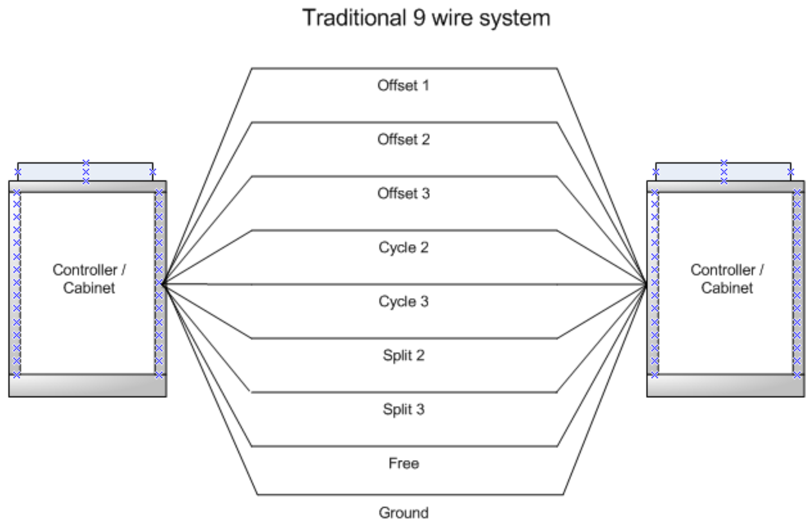
The Plan column corresponds to the "Naztec" Plan Number. Which is selected based on Plan Input A, Plan Input B, Plan Input C or Plan Input D as shown in the table described in section 3.2. The Offset column corresponds to the Offset input 1, Offset Input 2 or Offset Input 3. The coordination pattern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

4.3.3 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.

4.4 TS1/TS2 Controllers with Legacy Firmware Considerations

Legacy TS1 and/or TS2 Type 2 (NEMA) controllers utilize input pins to externally select plans and patterns. Many times they utilize a nine wire setup as shown below.



4.4.1 External Plan Setup

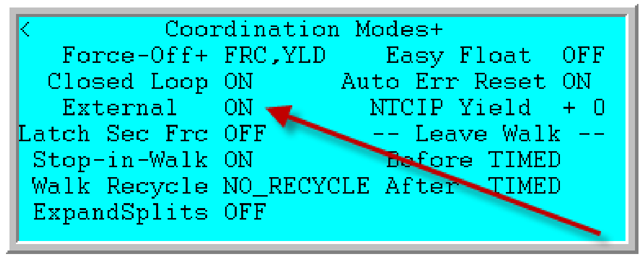
The NEMA TS2 specification, which Cubic | Trafficware adheres to have set up the four Plan inputs (Plan A, Plan B, Plan C and Plan D) to choose a NEMA Plan number (0-15) as shown in the table below. Cubic | Trafficware uses a slightly different set of Plan numbers for simplification of External I/O programming:

Original Naztec Plan #	NEMA Plan #	Cycle 3 Input	Cycle 2 Input	Split 3 Input	Split 2 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

If no cycle input is on, then the software assumes that the default cycle, Cycle 1, is active. Again, the synchronization (sync) pulse is generated when the offset input is turned off for a short period of time (typically 1 second).

4.4.2 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. In this case we have chosen 3 patterns (1,2,and 3) .

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

The Plan column corresponds to the "Naztec" Plan Number, which is selected based on Cycle 3 Input, Cycle 2 Input, Split 3 Input, and Split 2 Input as shown in the table described in section 3.4.1. 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. In the table above Pattern 1 is chosen for an active Cycle 1 Offset 1 input (Cycle 1 is the default cycle). Pattern 2 is chosen for active Cycle 2, Offset 1 inputs and Pattern 3 is chosen for an active Cycle 3 Offset 1 inputs.