

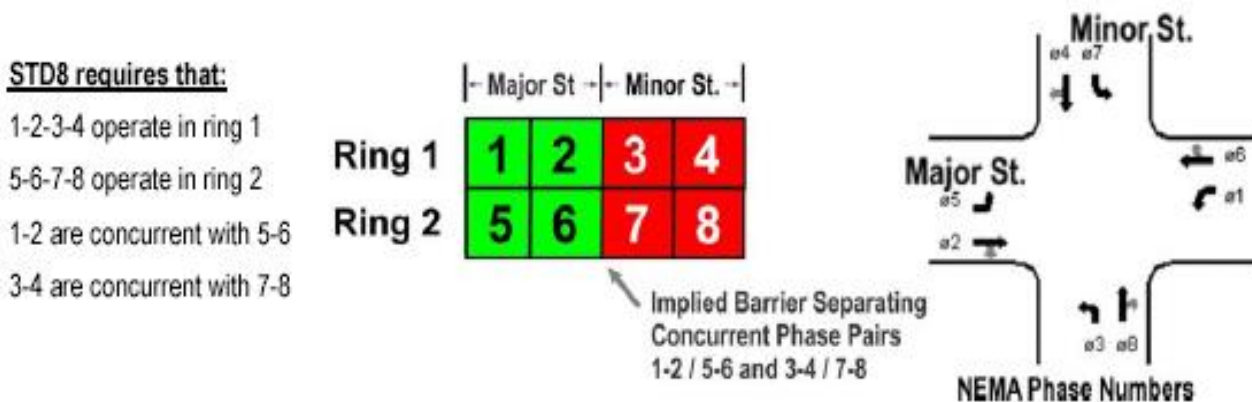
TecNote 1201 – V80.x Ring and Concurrency Programming considerations

The purpose of this TechNote is to assist the user in correctly programming ring and concurrency structures using V80.x software on Naztec ATC or 2070 Controller Platforms.

Version 80 supports thirty-two phases assigned to eight rings on Linux platforms and 32 phases assigned to six rings on OS9 platforms. Phases may time concurrently with phases in other rings that are defined as concurrent phases. Any phase not defined as a concurrent phase is considered to be a conflicting phase. The controller uses ring sequence and concurrency definitions to determine the order that the phases are serviced and to insure that conflicting phases never time concurrently. Phase concurrency establishes “barriers” between non-concurrent phases.

Ring and Concurrency overview using STD8

Many intersections utilize Standard 8 phase operation as shown on the diagram below



There are basic rules that controller software adheres to under standard eight phase quad operation.

- 1) Phases are separated into 2 sequential rings. Ring 1 runs phases 1-2-3-4 in sequential order and Ring 2 runs Phases 5-6-7-8 in sequential order. A phase can be skipped if there is no demand but the software will always follow the sequential order looking for demand.
- 2) Phases can be run in both rings at the same time as long as they are concurrent. Under standard 8 Phase operation Phases 1 and 2 are concurrent with phases 5 and 6 (main road) and phases 3 and 4 are concurrent with phases 7 and 8 (side road). A virtual barrier, defined by the concurrency table, separates the roads.
- 3) Phases in both rings must cross the barrier together before serving the next phase in sequence.

Programming the Sequence and Concurrency Tables

The screens below were programmed for STD8 operation as shown above. To get STD8 Operation, program the **Phase Mode** setting under **MM->1->2->1** to **STD8**.

There are two sets of menus in the controller to program the sequence and the concurrency. Both must be programmed to truly define the sequence and concurrency that is needed for a particular intersection. To program the sequence table chose the sequence menu at to **MM->->2->4**.

Ring Sequences	
1.Phases 1-16	2.Phases 17-32
First 16 Phases in Sequence	Second 16 Phases in Sequence

The sequence menu is designed to display 8 rings. Each ring defines the sequential operation of phases. Once a phase is selected for a ring, it cannot be used in another ring.

Menu selection 1 displays eight rings that can utilize first 16 phases in the sequence

Sequence of Phases									
Seq#	Ring#	<..1..2..3..4..5..6..7..8>							
1	1	1	2	3	4	0	0	0	0
1	2	5	6	7	8	0	0	0	0
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0
1	8	0	0	0	0	0	0	0	0

Sequence of Phases									
Seq#	Ring#	<..9..10..11..12..13..14..15..16>							
1	1	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0
1	8	0	0	0	0	0	0	0	0

Menu selection 2 displays an additional 16 phases in the sequence

Sequence of Phases									
Seq#	Ring#	<..17..18..19..20..21..22..23..24>							
1	1	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0
1	8	0	0	0	0	0	0	0	0

Sequence of Phases									
Seq#	Ring#	<..25..26..27..28..29..30..31..32>							
1	1	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0
1	8	0	0	0	0	0	0	0	0

To program the concurrency tables go to **MM->1->1->4**

Ring, Start, Concurrence	
1.Phases 1-16	2.Phases 17-32
First 16 Phases that can be concurrent	Second 16 Phases that can be

Menu selection 1 will display a row for each Phase and the first 16 phases that can be programmed to be concurrent to that phase. You must also program the ring number for that phase as programmed in the sequence table. Finally you can program a startup color for the associated channel.

Menu selection 2 will display a row for each Phase and the next 16 phases that can be programmed to be concurrent to that phase. You must also program the ring number for that phase as programmed in the sequence table. Finally you can program a startup color for the associated channel.

```
< Concurrent Phases >
P Ring StartUp ..1..2..3..4..5..6..7..8
1 1 RED 5 6 0 0 0 0 0 0 0
2 1 RED 5 6 0 0 0 0 0 0 0
3 1 RED 7 8 0 0 0 0 0 0 0
4 1 RED 7 8 0 0 0 0 0 0 0
5 2 RED 1 2 0 0 0 0 0 0 0
6 2 RED 1 2 0 0 0 0 0 0 0
7 2 RED 3 4 0 0 0 0 0 0 0
8 2 RED 3 4 0 0 0 0 0 0 0
9 0 RED 0 0 0 0 0 0 0 0 0
10 0 RED 0 0 0 0 0 0 0 0 0
11 + 0 RED 0 0 0 0 0 0 0 0 0
```

```
< Concurrent Phases >
P Ring StartUp .17.18.19.20.21.22.23.24
1 1 RED 0 0 0 0 0 0 0 0 0
2 1 RED 0 0 0 0 0 0 0 0 0
3 1 RED 0 0 0 0 0 0 0 0 0
4 1 RED 0 0 0 0 0 0 0 0 0
5 2 RED 0 0 0 0 0 0 0 0 0
6 2 RED 0 0 0 0 0 0 0 0 0
7 2 RED 0 0 0 0 0 0 0 0 0
8 2 RED 0 0 0 0 0 0 0 0 0
9 0 RED 0 0 0 0 0 0 0 0 0
10 0 RED 0 0 0 0 0 0 0 0 0
11 + 0 RED 0 0 0 0 0 0 0 0 0
```

```
< Concurrent Phases >
P Ring StartUp ..1..2..3..4..5..6..7..8
12 - 0 RED 0 0 0 0 0 0 0 0 0
13 0 RED 0 0 0 0 0 0 0 0 0
14 0 RED 0 0 0 0 0 0 0 0 0
15 0 RED 0 0 0 0 0 0 0 0 0
16 0 RED 0 0 0 0 0 0 0 0 0
17 0 RED 0 0 0 0 0 0 0 0 0
18 0 RED 0 0 0 0 0 0 0 0 0
19 0 RED 0 0 0 0 0 0 0 0 0
20 0 RED 0 0 0 0 0 0 0 0 0
21 0 RED 0 0 0 0 0 0 0 0 0
22 + 0 RED 0 0 0 0 0 0 0 0 0
```

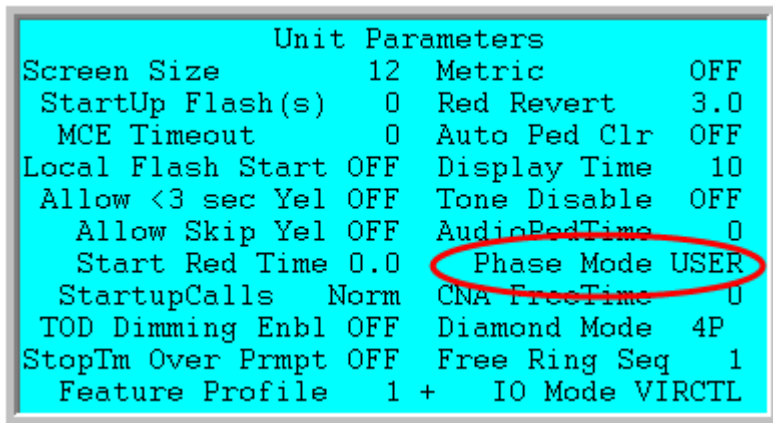
```
< Concurrent Phases >
P Ring StartUp .17.18.19.20.21.22.23.24
12 - 0 RED 0 0 0 0 0 0 0 0 0
13 0 RED 0 0 0 0 0 0 0 0 0
14 0 RED 0 0 0 0 0 0 0 0 0
15 0 RED 0 0 0 0 0 0 0 0 0
16 0 RED 0 0 0 0 0 0 0 0 0
17 0 RED 0 0 0 0 0 0 0 0 0
18 0 RED 0 0 0 0 0 0 0 0 0
19 0 RED 0 0 0 0 0 0 0 0 0
20 0 RED 0 0 0 0 0 0 0 0 0
21 0 RED 0 0 0 0 0 0 0 0 0
22 + 0 RED 0 0 0 0 0 0 0 0 0
```

```
< Concurrent Phases >
P Ring StartUp ..1..2..3..4..5..6..7..8
23 - 0 RED 0 0 0 0 0 0 0 0 0
24 0 RED 0 0 0 0 0 0 0 0 0
25 0 RED 0 0 0 0 0 0 0 0 0
26 0 RED 0 0 0 0 0 0 0 0 0
27 0 RED 0 0 0 0 0 0 0 0 0
28 0 RED 0 0 0 0 0 0 0 0 0
29 0 RED 0 0 0 0 0 0 0 0 0
30 0 RED 0 0 0 0 0 0 0 0 0
31 0 RED 0 0 0 0 0 0 0 0 0
32 0 RED 0 0 0 0 0 0 0 0 0
```

```
< Concurrent Phases >
P Ring StartUp .17.18.19.20.21.22.23.24
23 - 0 RED 0 0 0 0 0 0 0 0 0
24 0 RED 0 0 0 0 0 0 0 0 0
25 0 RED 0 0 0 0 0 0 0 0 0
26 0 RED 0 0 0 0 0 0 0 0 0
27 0 RED 0 0 0 0 0 0 0 0 0
28 0 RED 0 0 0 0 0 0 0 0 0
29 0 RED 0 0 0 0 0 0 0 0 0
30 0 RED 0 0 0 0 0 0 0 0 0
31 0 RED 0 0 0 0 0 0 0 0 0
32 0 RED 0 0 0 0 0 0 0 0 0
```

User defined Sequences and Concurrency Rules

The user has the ability to define sequences and concurrencies to accommodate various traffic geometries by selecting the unit parameter known as **Phase Mode to User** under **MM-1-2-1** as shown on the screen below. Phase mode must be set to user before programming the phase sequence and concurrency screens.



Unit Parameters			
Screen Size	12	Metric	OFF
StartUp Flash(s)	0	Red Revert	3.0
MCE Timeout	0	Auto Ped Clr	OFF
Local Flash Start	OFF	Display Time	10
Allow <3 sec Yel	OFF	Tone Disable	OFF
Allow Skip Yel	OFF	AudioPedTime	0
Start Red Time	0.0	Phase Mode	USER
StartupCalls	Norm	CNA FreeTime	0
TOD Dimming Enbl	OFF	Diamond Mode	4P
StopTm Over Prmpt	OFF	Free Ring Seq	1
Feature Profile	1 +	IO Mode	VIRCTL

When modifying phase sequences and concurrencies, V80 was updated to accommodate various iterations of sequencing and concurrency needs. Its programming has been modified and will act differently than prior versions of Trafficware controller software (V61.x, V65.x and V76.x) . For V80.x the following rules must be adhered when programming the sequence and concurrency structure to properly run the controller for the desired operation declared by the agency.

1. The goal of the V80.x ring logic is to never skip a phase. To achieve this operation, the precedent of a phase is **first by it ring** and **second by its position in the sequence**.
2. There **must always** be a phase concurrent at the first barrier (add dummy phase “0” if necessary).
3. There **must always** be a phase assigned in ring 1 for each concurrency barrier.
4. **Use dummy phase “0’s” to align the barriers.** Phases assigned to each column in the sequence table must be concurrent with each other. Therefore with V80.x the position of the phase in the sequence is needed to give the controller some information about how to prioritize. In versions prior to V80.x this was not necessary.

These rules will address the prioritization aspects of V80.x sequence concurrency algorithm, which implies you need to place dummy phase “0” in front of a phase, if you want the one in the other ring to service it first in sequence when the concurrent phases are part of two different barrier groups.

NOTE: The concurrency table programming (MM-1->4->1) works the same as prior versions of software but the user must program the sequence tables differently in V80.x

To understand this new algorithm, consider the following structure that is required by the agency:

Here's the problem. Imagine the following ring & barrier structure:

Ring 1	Φ 1	Φ 2	Φ 10	Φ 11	Φ 20	Φ 21	Φ 3	Φ 4
Ring 2	Φ 5	Φ 6					Φ 7	Φ 8

In controller software versions prior to V80.x, the sequence table would have been programmed as shown below

Sequence of Phases									
Seq#	Ring#	<..1..2..3..4..5..6..7..8>							
1	1	1	2	10	11	20	21	3	4
1	2	5	6	7	8	0	0	0	0
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0

Now imagine that the controller is crossing the barrier at the end of phases 2 and 6. The controller software must decide if it is going to service phase 10 or phase 7, because they're not compatible. If there is a call on both phase 10 and 7, it will serve phase 10 because it is in ring 1. If there are calls on phases 11 and 7 (no call on 10), the controller will service phase 7 because it is before 11 in the sequence. As you can see, this could lead to skipping phases 11, 20, and 21. In version 80 we added the ability to eliminate this issue by adding place holders("0") to mimic the concurrency in the sequence table

The correct implementation for V80.x of the example above is:

Sequence of Phases									
Seq#	Ring#	<..1..2..3..4..5..6..7..8>							
1	1	1	2	10	11	20	21	3	4
1	2	5	6	0	0	0	0	7	8
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0

The requirement about having a phase in every barrier for ring 1 may not be immediately obvious from the example above, but if you do not follow this rule there may be problems with coordination. Specifically, there may be problems with determining force-off points correctly.

As another example consider the sequence and concurrency requirements shown below:

Ring 1	Φ 2		Φ4		Φ 3	Φ 1
Ring 2	Φ 7	Φ 5	Φ 15	Φ 6	Φ 16	Φ 18

In controller software versions prior to V80.x, the sequence table would have been programmed as shown below

Sequence of Phases									
Seq#	Ring#	<..1..2..3..4..5..6..7..8>							
1	1	2	4	3	1	0	0	0	0
1	2	7	5	15	6	16	18	7	8
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0

Software prior to V80.x, when running phases 2 and 5 with demand on Phases 3 and 6, phase 6 could be skipped. This was because phase 3 in the ring sequence has a higher “priority” than phase 6 because not only is in a lower number ring, but it is also earlier in sequence.

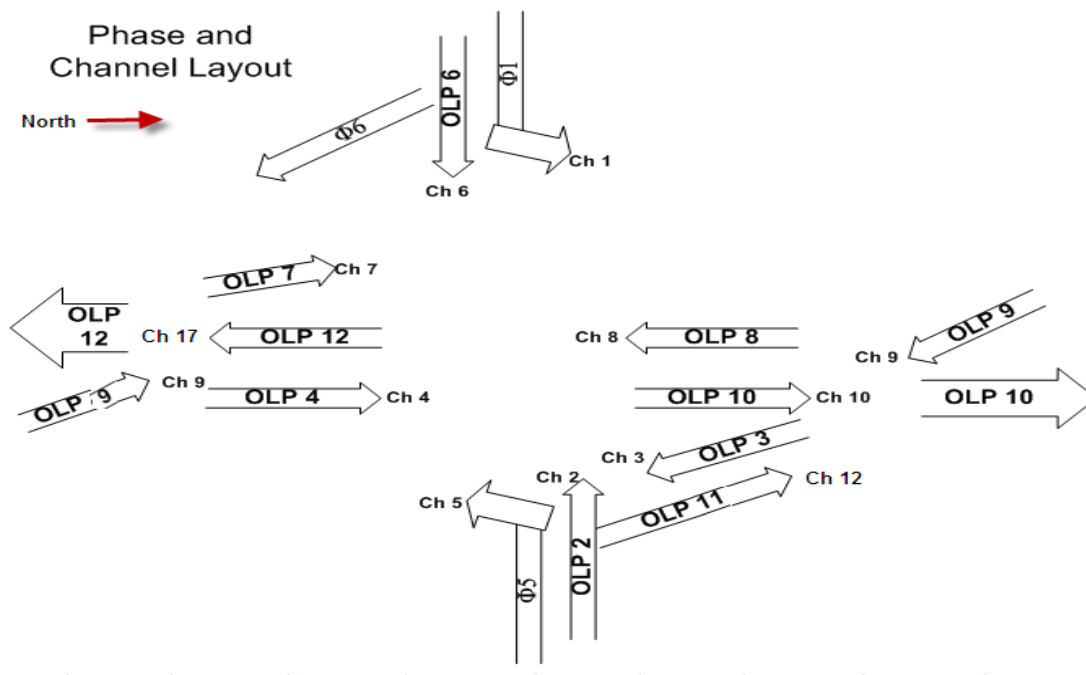
In V80.x to get the desired operation, please program the Ring sequence table as shown below:

Sequence of Phases									
Seq#	Ring#	<..1..2..3..4..5..6..7..8>							
1	1	2	0	4	0	3	1	0	0
1	2	7	5	15	6	16	18	7	8
1	3	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0

This way, in the event that phase 4 does not have a call, but phase 3 does and phase 6 has a call, the controller software will service 6 before 3.

NOTE: Dual Entry (MM->1->1->2) should **NOT** be set on any phases that are a part of a barrier which is not fully concurrent. The reason is because the Dual Entry programming checks to see if the phase that is next is compatible with the dual-entry phase using the assumption that the software is crossing a barrier.

As a final example consider the CFI Intersection shown below:



The operational requirements are as shown below:

Ring 1	Φ 1	Φ 2	Φ 9	Φ 10	Φ 11	Φ 3	Φ 4
Ring 2	Φ 5	Φ 6	Φ 14	(dummy)		Φ 7	Φ 8
Ring 3	Φ 13	(dummy)	Φ 15	Φ 16	ped	Φ 12	(ped 4&8)

The proper programming in V80.x for the ring sequence table will be:

Sequence of Phases									
Seq#	Ring#	<..1..2..3..4..5..6..7..8>							
1	1	1	2	9	10	11	3	4	0
1	2	5	6	14	0	0	7	8	0
1	3	13	0	15	16	0	12	0	0

Summary

In V80.x, the expanded ability of programming 32 phases, eight rings and up to 31 concurrent phases gives the agency flexibility of accommodating various geometries. In particular, the programming of sequence tables are different than prior versions of Trafficware's software. As always, the agency should fully test the specialized programming to insure the safety of the motoring public prior to installing in the field.