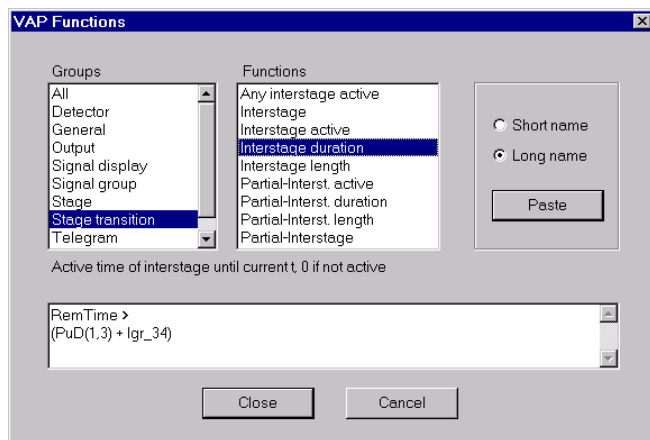


B.5 Funkcje VAP

W odróżnieniu od standardowych programów graficznych, w VisVAP wszystkie dostępne funkcje i rozkazy VAP mogą być wybierane z okna dialogowego i użyte w warunkach, instrukcjach i wyrażeniach. Dialog może być uruchomiony z poziomu każdego warunku lub instrukcji na schemacie lub z drugiej kolumny okna wyrażen przez użycie

- menu kontekstowego (prawy przycisk myszy) lub
- Shift + F2 lub
- EDIT - VAP-FUNCTIONS...



Funkcje VAP są w spisie pogrupowane. Po wybraniu grupy, wszystkie odpowiednie funkcje VAP są pokazane w oknie spisu. Po wybraniu listy funkcji, w oknie poniżej jest pokazywana krótka informacja o każdej z nich. Wybrana funkcja może być wprowadzona w miejsce aktualnej pozycji kursora w edytowanym oknie przy użyciu przycisku 'Paste'. Możliwe jest użycie długiej lub krótkiej nazwy funkcji. Gdy użyto 'Paste' VisVAP wprowadza funkcję o ścisłej syntaktyce VAP. Wymagane jest tylko uzupełnienie argumentów funkcji. Możliwe jest także wprowadzenie więcej niż jednej funkcji w edytowanym wierszu i połączenie ich przy użyciu operatorów (patrz B.3). Dla doprowadzeń linii wewnątrz symbolu po prostu wprowadź Ctrl+ENTER na

odpowiednich pozycjach wewnątrz wiersza edycji. Po zakończeniu dialogu zawartość wiersza edycji jest kopiowana jako tekst w symbolu.

Poniżej znajduje się kompletna lista funkcji, rozkazów i parametrów dostępnych w aktualnej wersji VAP-u. Jest tu wybór długich i krótkich syntaktyk funkcji. Dla czytelności została opuszczona lista argumentów dla krótkich nazw. Są one identyczne jak dla długich nazw.

Długa nazwa	Krótką	Znaczenie
Any_interstage_active	Aia	Zwraca 1 jeśli jakiś międzystan jest aktywny, inaczej 0.
Clear_front_ends (<nr>)	Cfe	Kasuje liczbę wykrytych pojazdów, które najechały na detektor <nr>
Clear_rear_ends (<nr>)	Cre	Kasuje liczbę wykrytych pojazdów, które zjechały z detektora <nr>
Current_state (<nr>, <stan>)	Cst	Zwraca 1 gdy bieżący stan grupy sygnalizacyjnej <gs_no> jest <stan>, gdy nie to zwraca 0.
Cycle_second	T	Zwraca aktualną sekundę cyklu.
Cycle_second_between (<początek>, <koniec>, <typ operatora>)	Csb	zwraca 1 gdy bieżąca sekunda cyklu jest pomiędzy <początek> (p) i <koniec>, w innym przypadku zwraca 0. <typ operatora> określa dokładnie znaczenie „pomiędzy”: 0: $p < t < k$; 1: $p < t \leq k$; 2: $p \leq t < k$; 3: $p \leq t \leq k$.
Cycle_time	Tc	Zwraca aktualną długość cyklu.
Desired_state (<nr>, <stan>)	Dst	Zwraca 1 gdy ostatni pożądaný stan (włączony przez Interstage lub Set_sg) grupy sygnalizacyjnej <gs_no> jest <stan>, gdy nie to zwraca 0.
Detection (<nr>)	Det	zwraca 1 gdy impulsowa pamięć detektora <det_nr> jest 1 (tj. wykryty został pojazd najeżdżający na detektor) lub pojazd znajduje się na detektorze. W pozostałych przypadkach zwraca 0.
Front_ends (<nr>)	F_e	zwraca liczbę wykrytych pojazdów, które najechały na detektor <det_nr>, czyli jest to liczba pojazdów które najechały na detektor od ostatniego wywołania clear_front_ends.
Headway(<nr>)	Hdw	Zwraca odstęp detektora <nr> od ostatniego wzbudzenia (aktywacji).
In_Frame(<nr>)	InFr	Zwraca czas, jaki upłynął od startu ramki linii <nr> w aktualnym planie ramowym (odpowiednio do aktualnie aktywnego numeru programu).
Intergreen(<gs1>, <gs2>)	IntG	Zwraca czas międzycielony między grupami sygnalizacyjnymi <gs1> i <gs2> jak definiuje zbiór *.pua.
Interstage(<fazal>, <faza2>)	Is	Uruchamia międzfazę

Długa nazwa	Krótką	Znaczenie
Interstage_active(<fazal>, <faza2>)	IsA	Zwraca 1 jeśli międzyfazę jest aktywny, inaczej 0.
Interstage_duration(<fazal>, <faza2>)	IsT	Zwraca aktualną sekundę międzyfazy lub 0 w wypadku, gdy międzyfaza nie jest aktywny
Interstage_length(<fazal>, <stage2>)	IsL	Zwraca długość międzyfazy.
Marker_get(<nr>)	MGet	Zwraca wartość z kanału wejściowego <nr> .
Marker_put(<nr>, <wartość>)	MPut	Wstawia <wartość> do kanału wyjściowego <nr>.
Notrace	Notrace	Zatrzymuje tryb śledzenia.
Occup_rate(<nr>)	OccR	Zwraca wygładzony stopień zajęcia detektora <nr> [0..1].
Occupancy(<nr>)	OccT	Zwraca czas, jaki upłynął od wzbudzenia (aktywacji) detektora <nr> lub 0 jeśli brak obecności pojazdu na końcu kroku czasowego.
P_interstage(<nr>, <sek>)	Pis	Wywołuje międzyfazę <nr>, rozpoczynając od sekundy <sek> (zwykle 0).
P_interstage_active(<nr>)	Pisa	Zwraca 1 jeśli międzyfaza <nr> jest aktywny, inaczej 0.
P_interstage_duration(<nr>)	Pist	Zwraca aktualną sekundę międzyfazy <nr> zgodnie z jego definicją w zbiorze *.PUA .
P_interstage_length(<nr>)	Pisl	Zwraca statyczną długość międzyfazy <nr> jak zdefiniowano w zbiorze *.PUA.
Parameter (<nr>)	Par	Zwraca wartość parametru <nr> według definicji w pliku *.PUA (dla aktywnego programu sygnalizacji)
Presence(<nr>)	Call	Zwraca 1, jeśli detektor <nr> był wzbudzony od ostatniego sprawdzania. W następnym kroku czasowym będzie on automatycznie zerowany.
Prog_active	PAct	Zwraca nr aktywnego programu wybranego w VISSIM-ie [1..16]
Rear_ends (<nr>)	R_e	Zwraca liczbę wykrytych pojazdów, których tył opuścił detektor <det_nr>, czyli jest to liczba pojazdów które opuściły detektor od ostatniego wywołania clear_rear_ends.
Record_value(<nr>, <wartość>)	RecVal	Wysyła <wartość> do VISSIM-a dla wyjścia w Raporcie SR/detektor w "kolumnie nr <nr>"
Remaining_intergreen(<nr>)	Rig	Zwraca czas w sekundach do chwili, gdy grupa sygnalizacyjna <nr> może dostać zielone z macierzą czasów międzyzielonych.
Reset(<licznik czasu>)	Rst	Ustawia licznik czasu <licznik czasu> na 0.
Set_cycle_second(<sek>)	SetT	Ustaw aktualną sekundę cyklu na <sek> [1...999]
Set_des_speed (<desSpDecNo>, <vehClassNo>, <desSpDistrNo>)	Sdsp	przydziela pożądaną rozkład prędkości <desSpDistrNo> klasie pojazdów

Długa nazwa	Krótką	Znaczenie
<vehClassNo>, <desSpDistrNo>)		<vehClassNo> dla decyzji o prędkości regulowanej <desSpDistrNo>.
Set_prog(<nr>)	Sprg	Ustaw nr aktywnego programu na <nr> [1..16].
Set_route (<routDecNo>, <routeNo>, <flow>)	Srou	przydziela potok względny trasie <routeNo> z wyboru trasy <routDecNo> w bieżącym przedziale czasu (lub w kolejnym, gdy przedział nie jest obecnie aktywny) strumieniowi <flow>
Set_sg(<nr>, <światło>)	Ssg	Przełącz grupę sygnalizacyjną <nr> na <światło> uwzględniając wszystkie przejścia. <światło> może mieć jedną z następujących wartości: <div style="display: flex; justify-content: space-between;"> <div>czerwone</div> <div>żółte</div> <div>zielone</div> </div> <div style="display: flex; justify-content: space-between;"> <div>czerwone_m</div> <div>żółte_m</div> <div>zielone m</div> </div> <div style="display: flex; justify-content: space-between;"> <div>czerwonożółte</div> <div>zielonożółte</div> <div>czerwonozielone_m</div> </div> <div style="display: flex; justify-content: space-between;"> <div>wył_czerwone</div> <div></div> <div>ne_m</div> </div> <div style="display: flex; justify-content: space-between;"> <div>wył</div> <div></div> <div></div> </div>
Set_sg_direct(<nr>, <światło>)	Ssgd	Przełącza grupę sygnalizacyjną <nr> natychmiast na <światło> bez żadnych przejść i lekceważąc minimalne czasy świateł.
Sg_green(<nr>)	SetG	Przełącz grupę sygnalizacyjną <nr> na zielone.
Sg_red(<nr>)	SetR	Przełącz grupę sygnalizacyjną <nr> na czerwone.
Stage_active(<faza>)	StgA	Zwraca 1, jeśli faza <faza> jest aktywna, inaczej 0.
Stage_duration(<faza>)	StgT	Zwraca czas, jaki trwa faza <faza> lub 0 jeśli faza nie jest aktywna.
Start(<licznik czasu>)	-	Uruchamia licznik czasu <licznik czasu>.
Start_at(<licznik czasu>, <wartość>)	-	Uruchamia licznik czasu <licznik czasu> z wartością początkową <wartość>. Pierwszy przyrost czasu nastąpi w następnym kroku symulacji.
Stop(<licznik czasu>)	-	Zatrzymuje licznik czasu <licznik czasu> bez jego zerowania.
T_free (<nr>)	Tf	Zwraca czas, jaki upłynął od końca czerwonego grupy sygnalizacyjnej <nr> (zielone plus czas czerwonożółtego).
T_green(<nr>)	Tg	Zwraca jaki upłynął czas światła zielonego grupy sygnalizacyjnej <nr> lub 0 jeśli grupa sygnalizacyjna nie jest zielona.
T_green_min(<nr>)	Tgmin	Zwraca minimalny czas światła zielonego grupy sygnalizacyjnej <nr>.
T_red(<nr>)	Tr	Zwraca jaki upłynął czas światła czerwonego grupy sygnalizacyjnej <nr> lub 0 jeśli grupa sygnalizacyjna nie jest czerwona.
T_red_min(<nr>)	Trmin	Zwraca minimalny czas światła czerwonego grupy sygnalizacyjnej <nr>.

Długa nazwa	Krótką	Znaczenie
T_stop(<nr>)	Ts	Zwraca czas jaki upłynął od końca światła zielonego grupy sygnalizacyjnej <nr> (czas czerwonego plus czas żółtego).
Tele_calling_point(<nr>)	TeCpt	Zwraca numer punktu zgłoszenia z telegramu numer <nr>.
Tele_count	TeCnt	Zwraca liczbę odebranych telegramów w aktualnym kroku symulacji.
Tele_course(<nr>)	TeCrs	Zwraca numer kursu z telegramu numer <nr>.
Tele_delay(<nr>)	TeDel	Zwraca stratę czasu w sekundach z telegramu numer <nr>.
Tele_line(<nr>)	TeLin	Zwraca numer linii z telegramu numer <nr>.
Tele_man_direction(<nr>)	TeDir	Zwraca kod ręcznego kierunku z telegramu numer <nr>.
Tele_passengers (<nr>)	TePas	Zwraca liczbę pasażerów z telegramu komunikacji zbiorowej numer <nr>.
Tele_prio(<nr>)	TePri	Zwraca wartość priorytetu z telegramu numer <nr>.
Tele_route(<nr>)	TeRou	Zwraca numer trasy z telegramu numer <nr>.
Tele_tram_length(<nr>)	TeLng	Zwraca długość tramwaju z telegramu numer <nr>.
Test_intergreen (<nr>)	Tig	Zwraca 1 gdy wszystkie czasy międzyszielone względem grupy sygnalizacyjnej <nr> są zakończone przed przejściem grupy na światło zielone.
Trace(all)	-	Zapisuje rozkazy programu i wartości wszystkich zmiennych w pliku śledzenia.
Trace(command)	-	Zapisuje rozkazy programu w pliku śledzenia.
Trace(variable)	-	Zapisuje wartości wszystkich zmiennych w pliku śledzenia.
Trace(variable(<var>))	-	Zapisuje wartość zmiennej <var> w pliku śledzenia.
Trace(variable(<var1>, <var2>))	-	Zapisuje wartości zmiennych <var1> i <var2> w pliku śledzenia.
Veh_length (<nr>)	V_1	Zwraca długość ostatniego pojazdu, który opuścił detektor <nr> podczas bieżącej sekundy symulacji.
Velocity(<nr>)	V	Zwraca prędkość w m/s ostatniego pojazdu, który był wykryty przez detektor <nr> w aktualnym kroku czasu lub 0 jeśli nie wykryto żadnego pojazdu w tym kroku czasu.