

Delft University of Technology

### A Software Prototype for Isolated Ramp-Metering

Taale, Henk

**Publication date** 1991 **Document Version** Final published version

**Citation (APA)** Taale, H. (1991). A Software Prototype for Isolated Ramp-Metering. Rijkswaterstaat Dienst Verkeerskunde.

Important note To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright** Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology. For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

# **DRIVE PROJECT (V 1035)**

# CHRISTIANE

(MOTORWAY TRAFFIC FLOW MONITORING AND CONTROL)

to be included in DELIVERABLE number 7b

A software prototype for isolated ramp-metering

CXR92014.rap

- 1 July 1991 -

# DRIVE PROJECT (V 1035)

# CHRISTIANE

(MOTORWAY TRAFFIC FLOW MONITORING AND CONTROL)

to be included in DELIVERABLE number 7b

A software prototype for isolated ramp-metering

CXR92014.rap

- 1 July 1991 -

Project Reference Number: V 1035

Project Acronym:	CHRISTIANE
Project Title:	MOTORWAY TRAFFIC FLOW MONITORING AND CONTROL
Prime Contractor:	Institut National de Recherche sur les Transports et leur Sécurité (INRETS, France)
List of Partners:	Scetauroute (France) Technical University of München (Germany) Transport and Road Research Laboratory (United Kingdom) Wootton Jeffrey's Consultants (United Kingdom) University of Thessaloniki (Greece) Rijkswaterstaat (The Netherlands)
Subcontractors:	CERT/DERA/SERALP (France)
Document status:	public report
Prepared by:	Dutch Ministry of Transport and Public Works Transportation and Traffic Research Division P.O.box 200 3000 BA Rotterdam tel: +31-10-4026610 fax: +31-10-4148115
Authors:	ir. Henk Taale ir. Frans Middelham
Date:	1 July 1991

# CONTENTS

1.	INTRODUCTION	2
	1.1. DRIVE-context	1
	1.2. Project-context	1
	1.3. Summary	2
	······································	
2	FLEXSYT	3
	21 Introduction	3
	2.2. Dosign requirements	3
	2.2. Design requirements	3
	2.0. MANAGEN CONCEPT	4
		4
		4
	2.5.1. Elements of FLEXCOL-/6	4
	2.5.1.1. Names of elements	4
	2.5.1.2. Indices of elements	5
	2.5.2. Events in FLEXCOL-76	5
	2.5.3. Conditions in FLEXCOL-76	5
	2.5.4. Functions in FLEXCOL-76-	6
	2.5.5. Collections in FLEXCOL-76-	6
	2.5.6 Behaviour of elements in FLEXCOL-76-	7
	2.5.7 Formula concept of ELEXCOL-76-	8
	2.5.7. Formula concept of r LEXCOL 70	0
	2.5.7.1. Event assignment statement	0
		8
	2.6. Structure of the program FLEXSY1-I-	9
	2.6.1. Input data sets of FLEXSYT-I-	10
	2.6.2. Output data sets of FLEXSYT-I	11
	2.7. Comparison of FLEXSYT-I- with real life	11
	2.8. Development of FLEXSYT-II-	11
3.	PROTOTYPE	13
0.	3.1 Introduction	13
	3.2 MANDAT detect	10
		10
		15
		15
	3.3.2. Counting and measuring part	17
	3.3.3. Signal handling part	20
	3.4. Conclusion	21
APF		23
A.	Examples	
- *		

-----

يند افكاركيسه ه

A.1.		24
A.2.	CONDAT dataset	25
A.3.	NETDAT dataset	26
A.4.	REGFIL dataset	28
A.5.	RESFIL dataset	29

В.	Prototyp	be input datasets	32
	B.1.	MANDAT dataset	32
	B.2.	CONDAT dataset	34

# 1. INTRODUCTION

# 1.1. DRIVE-context

One of the projects of the European programme DRIVE is concerned with the problem of "Motorway Traffic Flow Monitoring and Control" and is named *Christiane* (Ref.nr. V1035). Prime Contractor of the project is INRETS (F); the Partners are Scetauroute (F), Technical University Munich (FRG), TRRL (UK), Wootton Jeffrey's Consultants (UK), University of Thessaloniki (Greece) and Rijkswaterstaat (NL). Subcontractors are CERT, DERA and SERALP (F).

The general objective of DRIVE is threefold: improve road transport efficiency, improve road transport safety and reduce environmental pollution. Motorway traffic monitoring and control are related to all of these objectives. Motorway control aims at optimal utilisation of the capacity of motorway networks, for which motorway monitoring is a necessary prerequisite. With respect to control strategies the DRIVE objective is "to establish a common framework in which different system control methods can be accommodated without losing compatibility". (Annual Project Review Report of Christiane, 27-09-89).

# 1.2. Project-context

The type of motorway control considered in the Christiane project are control by Variable Message Signs (VMS) and by ramp-metering. The project is divided in two parallel developments: traffic flow modelling and traffic flow control. The models may be used for the monitoring purpose and also for designing or testing control strategies. In the control direction there are two approaches: aid-to-decision and automatic control.

The first step in the project has been a review of existing models and control strategies. The result of this review are reported in the project Deliverable no. 1 of Work Package 1 (April 1989). With respect to subject of the ramp-metering the review has recognised the following approaches:

- ALINEA, developed by the Technical University Munich, and tested by INRETS on the Boulevard Périphérique;
- a strategy developed by Wootton Jeffrey's, applied on several (6) ramps of the M6 in the UK;
- the strategies applied by Rijkswaterstaat at the Coentunnel and near Delft in The Netherlands;
- the theoretically developed strategy of the University of Thessaloniki.

A detailed comparison of the several strategies and a discussion on some main differences (like vehicle by vehicle vs. platoon metering, feedback vs. feedforward control etc.) may be found in the project Deliverable 5 of Work Package 3 called "Isolated Ramp-Metering" (April 1990).

# 1.3. Summary

It is the aim of the ramp-metering part of the project to use the existing expertise to identify the applicability of existing approaches. Apart from a desk-top comparison like the deliverable just cited, two **field trials** were planned:

- a field trial with ALINEA at the Coentunnel in Amsterdam, The Netherlands;
- a field trial with the Wootton Jeffrey's strategy on the BP in Paris, France.

These trials should lead to insight in the range of conditions over which the approaches are viable, the limits of the applicability of isolated control and possibly lead to control improvements. The results of the trial in The Netherlands are reported in Deliverable 7a: *Isolated Ramp-metering: Real life Study in the Netherlands*, march 1991.

The main conclusion of this report states that: The ALINEA algorithm for ramp-metering produces results comparable to or better than those of the RWS algorithm. ALINEA increases the total service of the system, the motorway and the ramp at the Coentunnel. When the homogenisation of the traffic flow in the bottleneck is one of the aims of ramp-metering, the fact that ALINEA gives a 10 % greater deviation in the speeds at the bottleneck is of importance.

For further research it is necessary to have a implementation of the RWS and ALINEA strategy in a software prototype. The theoretical background of both strategies is discussed in Deliverable 7a.

This document is a continuation of Deliverable 7a and discusses the implementation of the Rijkswaterstaat and ALINEA strategy in a software prototype. The prototype is developed in the computer program FLEXSYT. Therefore the program FLEXSYT is discussed first and then the software prototype of a ramp-metering installation is developed.

# 2. FLEXSYT

# 2.1. Introduction

**FLEX**ible-traffic-network-Simulation-studY-Tool was developed in the seventies en eighties by Frans Middelham. It is a computer program for traffic management studies and simulates traffic on a microscopic scale. On a stochastic base vehicles move through the network and travel times are calculated. In this way it is possible to do research on the structure of the network, such as the lay-out of intersections, length and number of lanes, effects of bus lanes, etc., and on traffic-control alternatives, like the fixed-time control strategy, vehicle-actuated control strategies, traffic-depended control strategies, etc. In this chapter the basic principles and the structure of FLEXSYT are discussed.

# 2.2. Design requirements

The design requirements for a traffic-control-language are of the same importance as those for a common computer language. When starting the work on FLEXSYT, the following design requirements had been formulated:

- well defined to be unambiguous;
- user friendly;
- no built-in control philosophy;
- independent of manufacturer's;
- independent of computer systems.

# 2.3. MANAGER concept

The MANAGER concept was chosen to avoid the use of a built-in control philosophy. It was found that, up to then, most of the traffic control simulation programs had a built-in control philosophy and therefore were problem dedicated. FLEXSYT, in contrary, is a general purpose traffic control simulation program.

The MANAGER of the program must specify some parts of the traffic control process at implementation time of the program. It gives the possibility to generalize signal handling and calculations. At the same time the MANAGER may specify a traffic control philosophy. It gives the MANAGER the opportunity to force its USERS to use a standard type of control

Examples of traffic control philosophy's that can be implemented are 'round-abouts', 'arterials', 'toll-plaza's', 'ramp-metering installations', etc.

# 2.4. USER concept

In FLEXSYT the USER is defined as the person to whom the MANAGER has allowed the use of the program and the use of his MANAGER dataset (MANDAT).

It's the task of the USER to solve a certain traffic control problem. Therefor the USER has to define his network and his problem dedicated control strategy.

# 2.5. Traffic control language (FLEXCOL-76-)

A traffic control language for program description of flexible controlled intersections was developed in 1976. Further extensions to meet the requirements of program descriptions of network control were added later. This flexible network traffic control language (known as FLEXCOL-76-) is based upon the rules of boolean algebra and the clear differentiation between the 'change of state of an element' and the 'state of an element'.

The 'change of state of an element' is called EVENT and has no logical meaning. The 'state of an element' has a logical meaning and is called CONDITION, that is: the description of the state of an element is logically true or false.

#### 2.5.1. Elements of FLEXCOL-76-

Basically the traffic control process has three (and no more than three) types of elements. Using these elements gives the opportunity to control the traffic in every desired manner. These elements are: MEMORY elements, TIMER elements and DETECTOR elements.

In many circumstances, the only values for a MEMORY element are '0' and '1'. For this reason a LOGICAL element has been created. Thanks to the simple contents of this element, the syntax of a LOGICAL element is simpler. This increases readability of the traffic control program and gives the opportunity to decrease the use of computer-memory.

#### 2.5.1.1. Names of elements

To meet the requirements of flexibility and independency, the name of each element is declared by the MANAGER. Each element name can have up to 20 significant characters. Legal characters are: capitals, letters and 'underscores'. So legal element names are:

- MEASURE_PERIOD_	(memory-element),
- GREEN_TIMER_	(timer-element),
- OCCUPANCY_DETECTOR_	(detector-element),
- ALINEA_	(logical-element).

To be user friendly, to increase readability and to decrease the size of a traffic-controlspecification, the MANAGER may select an abbreviation instead of the full element-name.

#### 2.5.1.2. Indices of elements

Each element in the traffic control program can have an index. The index of an element can have three levels. Therefore one has to consider, that the traffic engineer has to deal with three levels of control. These levels are: NETWORK level, INTERSECTION level and SIGNAL level.

When dealing for instance with intersection number 001 and with signal number 05 and 45, examples of legal combinations of element names and indices are:

- METERING_TIME_001/00	(INTERSECTION-level).
- ONRAMP_VOLUME_001/00	(INTERSECTION-level)
GREEN_001/05	(SIGNAL-level),
- CLEARANCE_TIME 001/45	(SIGNAL-level)

Because of the fact that, at installation time of the traffic control language, the MANAGER doesn't know the intersection numbers nor the signal numbers to be used, we have to define implicit indices, symbolized by the character '\$'. Thus legal combinations of element names and indices, to be used by the MANAGER, are:

- MAX_METERING_TIME_\$\$\$/00	(INTERSECTION-level).
YELLOW_\$\$\$/\$\$	(SIGNAL-level).

If it is obvious, the index of an element may be omitted. The use of the default index increases the readability of the control program.

#### 2.5.2. Events in FLEXCOL-76-

The manipulation of all elements is effected by the description of the 'change of state' (EVENT) of these elements by the MANAGER and/or the USER. The syntax of an EVENT (of an element called M with index i) is:

S(Mi = x)

The meaning of 'x' is explained later.

The manipulation of logical elements is the description of the 'change of state' of such an element from 'false' to 'true' (SET EVENT) or from 'true' to 'false' (END EVENT or RESET EVENT). The syntax of these EVENTS (of an element called L with index i) is:

SLi and ELi

#### 2.5.3. Conditions in FLEXCOL-76-

The syntax of a test on the 'state of an element' (CONDITION) appears in six ways. In words, this test can be read as: is the contents of an element (not) equal (or greater) (or less) than the contents of an arithmetical expression. As invert operator the letter 'N' was chosen. The syntax of the six possible CONDITIONS is:

Here also the syntax of a test on the 'state of a logical element' is simpler:

Li and LiN

#### 2.5.4. Functions in FLEXCOL-76-

The meaning of 'x' in the above definitions is more complicated. When changing the state of MEMORY elements or TIMER elements and when using the change of states of DETECTOR elements, the MANAGER and the USER can use functions in a way similar to computer languages. The syntax of the use of these functions is defined in the following way: x is a constant, x is the contents of an element or x is part of a FUNCTION(x,y,z,..). FUNCTIONS that can be used in FLEXCOL-76- are:

- ADD(x,y)for readability: x+y, - SUBTRACT(x,y) for readability: x-y, for readability: x\*y, - MULTIPLY(x,y) - DIVIDE(x,y) for readability: x/y, - POWER(x,y) for readability: x\*\*y, - SQRT(x) (square-root), - ALOG(x) (natural-logarithm), - LOG(x) (common-logarithm), - EXP(x) (exponential), (absolute-value), - ABS(x)- MAX(x,y,z,..) (maximum), - MIN(x,y,z,..) (minimum).

The definition of 'x' is recurrent, so nesting of the functions is allowed. Functions in FLEXSYT follow the common rules in computation. Apart form using indices, an example of the manipulation of elements is exponential smoothing, with FLOW\_OLD\_ the element containing the smoothed value, ALPHA\_ the element containing the smoothing factor and FLOW\_NEW the element containing the value of the measured period:

S(FLOW\_OLD\_016 = ALPHA\_ \* FLOW\_NEW\_016 + (1-ALPHA\_) \* FLOW\_OLD\_016)

#### 2.5.5. Collections in FLEXCOL-76-

Many times, the traffic engineer implicitly uses the collection or set concept. Examples of a collection in traffic control are: the intersections of a network, the signals of an intersection, the signals of a stage an the conflicts of a signal.

To suit the use of collections, the MANAGER must define the names of the collections he wants his USER to operate with. Definition of these names follows the same rules as the definition of the names of elements. Legal collection names are:

- METERING_INSTALL_	(NETWORK-level),
- METERING_SIGNAL_	(INTERSECTION-level).

page -6-

In collections the use of an index of an element is implicit. An example of the use of a collection is:

METERING\_SIGNAL\_001: 01 05 45

#### 2.5.6. Behaviour of elements in FLEXCOL-76-



# Figure 1. Logical behaviour of the MEMORY-, TIMER-, and DETECTOR-elements in time

The behaviour of the elements in time is given in figure 1. As can be seen from this figure, there is a difference in the use of the arrows:

- an arrow pointing *towards* the figure is a REACTION EVENT. The specification of the 'change of state' of an element is done by the MANAGER and/or the USER.
- an arrow pointing *outwards* the figure is an ACTION EVENT. The MANAGER and/or the USER only may use the 'change of state' but cannot cause the 'change of state'.

#### 2.5.7. Formula concept of FLEXCOL-76-

#### 2.5.7.1. Event assignment statement

The program of the traffic control application (being the specification of the signal plan) contains the descriptions of the 'change of states' of elements in EVENT ASSIGNMENT STATEMENTS. In general, the EVENT ASSIGNMENT STATEMENT is defined in the following way:

REACTION EVENT .= ACTION EVENT . 'condition'

So the manipulation of an element is caused by an event. The way in which this occurs is very similar to the way in which SUBROUTINE calls are effected in computer programs. In the above general description, the meaning of each part is:

- 'REACTION EVENT' contains the description of the element(s) to be manipulated,
- '.=' is the event assignment operator sign,
- 'ACTION EVENT' contains the description of element(s), the manipulation of which causes the assignment,
- '.' is an operator sign
- 'condition' contains the description of a combination of logical conditions that must be fulfilled to effect the assignment. The definition of 'condition' is recurrent, so:
  - 'condition' is 'empty',
  - 'condition' is CONDITION,
  - 'condition' is (CONDITION)N,
- 'condition' is CONDITION + CONDITION,
- 'condition' is CONDITION . CONDITION.

This combination of logical conditions follows the common rules of the boolean-algebra, with the operators:

- 'N' is the LOGICAL INVERT,
- '+' is the LOGICAL OR,
- '.' is the LOGICAL AND.

The definition of the EVENT ASSIGNMENT STATEMENT is also recurrent. So an ACTION EVENT may be substituted by a REACTION EVENT. This allows (but doesn't force) the MANAGER and USER to construct complex control algorithms.

2.5.7.2. Boolean assignment statement

There also exists an easier description of the EVENT ASSIGNMENTS on logical elements if the formulation of the SET EVENT is the inverse of the formulation of the RESET EVENT. For example (using the logical elements A, B and C with index i) when:

SAI .= SBI.CI + SCI.BI EAI .= EBI + ECI

then the simplified form of these two EVENT ASSIGNMENT STATEMENTS is one BOOLEAN ASSIGNMENT STATEMENT:

#### Ai = Bi . Ci

The BOOLEAN ASSIGNMENT STATEMENT is defined as:

'state of logical element' = 'condition'

Then meaning of each part is:

- 'state of logical element' contains the description of logical elements to be manipulated,
- '=' is the boolean assignment operator sign,
- 'condition' is similar to the definition in the event assignment statement, except that 'condition' may not be 'empty'.



# 2.6. Structure of the program FLEXSYT-I-

#### Figure 2. Structure of FLEXSYT

The structure of FLEXSYT is shown in figure 2. FLEXSYT consists of a number of programs which require three input data sets. In the MANDAT dataset, the names of

elements and collections are specified. Also the traffic control routines on network-, controller, and signal level are given here, being the default routines for the user.

In the CONDAT dataset, the problem defined input of the traffic control strategies are given.

In the NETDAT dataset, the problem defined input for the network lay-out is given, as well as the simulation parameters, the traffic flow values and the location of the stoplines and detectors.

Before the simulation starts, the input data sets are screened and checked for a correct syntax and use of parameters by the programs FLXMAN, FLXCON and FLXNET. With FLXDIS it is possible to match a picture of the network, drawn with Dr. Halo or Dr. Genius, with the control strategy specified in the CONDAT dataset. One can check this control strategy by hand, thus without running a simulation, with FLXCOL.

If the input has been compiled successfully, the simulation is effected with the program FLXSIM. With FLXMON the simulation can be followed on the screen. If 'congestion' occurs during simulation. a dataset DMPFIL is created with information about the state of the network. Errors in the traffic control strategy may be traced in the dataset TRCFIL, containing an 'event trace' and available upon request.

While testing the control strategy, a 'state trace' for every simulation second may be obtained with the help of the program FLXREG and available on the REGFIL dataset. This offers a very helpful facility during the development and testing of control strategies.

The results of the simulation can be obtained with FLXRES and are available in the RESFIL dataset.

All other files that are made, like the MANFIL, CONFIL, NETFIL, DISFIL, SIMFIL, CNTFIL, etc., are intermediate and have no meaning to the user.

#### 2.6.1. Input data sets of FLEXSYT-I-

Preceding the start of a simulation run, the program FLEXSYT needs to be loaded with the specification of the subject to be studied or the problem to be solved. This specification is split up in three input data sets:

- MANager DATaset (appendix A.1), with items: title, definition of element names, standard network control statements, standard intersection control statements and standard signal control statements,
- CONtrol DATaset (appendix A.2), with items: title, specific network control statements and specific intersection/signal control statements.
- NETwork DATaset (appendix A.2), with items: title, simulation parameters, generator element values (traffic load), network element description, with: length, saturation flow, signal relation, speed or travel time (per car type) and route parameters (per car type), signal/stopline positions and detector element positions,

These datasets are necessary to run a simulation with FLXSIM. If FLXMON or FLXCOL is used some other files are necessary. The picture of the area to be studied, drawn with Dr. Halo of Dr. Genius, is contained in a .PIC file and the colour pallet in a .PAL file. With FLXDIS the signal movements and detectors defined in the CONDAT dataset can be matched with the picture. This information is stored in the DISplay DATaset and the DISFIL. The DISDAT dataset can be used later on for a new session with FLXDIS. The DISFIL is, together with the CONFIL, the .PIC file and the .PAL file, input for FLXMON and FLXCOL.

#### 2.6.2. Output data sets of FLEXSYT-I-

Before actual simulation, the program checks the input data very thoroughly. It issues DISASTER, FATAL, ERROR or WARNING messages if needed. With correct input data sets the program starts producing its results. In an interactive environment the SIMULATION DATA may appear immediate on a terminal, with items: successful end of simulation or end of simulation, because of congestion. The last message occurs when: the control plan is erroneous, the control plan isn't yet optimized or the network cannot handle the traffic. After a successful end of simulation with the program FLXREG one can produce a REGFIL (appendix A.4), with every (model) second, the state of signals and detectors. The REGFIL is very helpful during the development of the signal plan.

The results of a simulation are produced with the program FLXRES. The RESFIL (appendix A.5) contains the following traffic data:

- for the signals (totals at stoplines): car flow in vehicles/hour, delay in seconds, green time in seconds, non-green time in seconds and cycle time in seconds.
- for the network (per car type)(totals for each element): flow in vehicles/hour, distance travelled in vehicles\*kilometres/hour, delay in vehicles\*hour/hour, time spent in vehicles\*hour/hour, stops in vehicles/hour and maximum queue in vehicles.
- for the network (per car type)(totals for the network): total distance travelled in vehicles\*kilometres/hour, total time spent in vehicles\*hour/hour, total delay in vehicles\*hour/hour, total stops in vehicles/hour and speed in kilometres/hour,
- traffic streams from generators: delay in seconds/vehicle and distribution of delay (classwidth 10 seconds).
- traffic streams from stoplines: delay in seconds/vehicle and distribution of delay (classwidth 10 seconds).

#### 2.7. Comparison of FLEXSYT-I- with real life

A scientific comparison of FLEXSYT-I- with real life never had been carried out. This was due to the fact that the development of the program wasn't a part of a survey project nor was sponsored by any company and because a new version of FLEXSYT is being developed. After some studies with FLEXSYT, which gave good results, the program was adopted by the Dutch Ministry of Transport.

### 2.8. Development of FLEXSYT-II-

The good results obtained with FLEXSYT-I- were the reason for the Dutch Ministry of Transport to start a project for the development of FLEXSYT-II-. The main purpose of the project was to integrate some models of the computer program SIGSIM. The main

improvements of FLEXSYT-II-, in comparison with FLEXSYT-I-, are going to be:

- from 2 to 7 types of traffic participants: private-cars, lorries, trucks, buses, tramcars, pedestrians, bicycles,
- route-vehicles,
- secondary-conflicts
- unsignalized intersections.

The main objective of the project however is the scientific validation of the networkaspects of the models. For SIGSIM the scientific validation of the intersection-aspects was already performed.

The development of FLEXSYT-II- is in its final stage.

# 3. **PROTOTYPE**

### 3.1. Introduction

In this chapter the input data sets of FLEXSYT for a prototype program for an isolated ramp-metering installation are discussed. First a piece of the input dataset is given and then some comments. The complete text, in one piece, of the MANDAT and CONDAT dataset is contained in appendices B1 and B2. This prototype can be seen in operation on the demo floppy diskette, available with this deliverable.

### 3.2. MANDAT dataset

//MANDAT-dataset for a PROTOTYPE of an isolated RAMP-METERING INSTALLATION

//DETector-element-names
\_ SPEED\_DETECTOR\_ LONG\_LOOP\_ OCCUPANCY\_DETECTOR\_ SHORT\_LOOP\_

GREEN TIMER       YELLOW TIMER       RED_TIMER         METERTING_TIMER       MEASURE_TIMER       RED_TIMER         //MEMory-element-names       PRINT         MIN_GREEN_TIME       MAX_GREEN_TIME         MIN_GREEN_TIME       MAX_GREEN_TIME         MIN_RED_TIME       MAX_YELLOW_TIME         MIN_RED_TIME       MAX_YELLOW_TIME         MIN_METERING_TIME       MAX_METERING_TIME         MEASURE_PERIOD       PERIOD         MEASURE_MET_TIME       CLEARANCE_TIME         FLOW_OLD       FLOW_CUR_         SET_ON_FLOM       SET_OFF_FLOM         SPEED_NEW       SPEED_OLD         VEH_SPEED_OLD       VEH_SPEED_CUR         SPEED_UPSTREAM       SPEED_OLD_SET_OFF_SPEED_         SET_ON_SPEED       SET_OFF_SPEED_         SET_ON_SPEED       SET_OFF_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOM       ALPHA_DECREASE_FLOM_BETA_DECREASE_SPEED_BETA_         ALPHA_       OCCUPANCY_SETPOINT_ALINEA_CONSTANT_ORAMP_VOLUME_         //LOGical-element-names	//TIMer-element-names		
METERING_TIMER_       MEASURE_TIMER_       NEASURE_TIMER_         MEMORY-element-names       PRINT_         MIN_GREEN_TIME_       MAX_GREEN_TIME_         MIN_RED_TIME_       MAX_YELLOW_TIME_         MIN_RED_TIME_       MAX_YELLOW_TIME_         MIN_RED_TIME_       MAX_YELLOW_TIME_         MIN_RED_TIME_       MAX_YELLOW_TIME_         MIN_RED_TIME_       MAX_METERING_TIME_         MEASURE_PERIOD_       PERIOD_         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         VEH_SPEED_OLD_       VEH_SPEED_CUR_       SPEED_CUR_         SPEED_UPSTREAM_       SPEED_DOLNESTREAM_       SPEED_CUR_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       ALPHA_DECREASE_FLOW_         BETA_INCREASE_FLOW_       ALPHA_DECREASE_SPEED_       BETA_         CAPACITY_       OCCUPANCY_SETPOINT_       ALPHA_         ALPHA_       NCREASE_SPEED_       BETA_       DK_         MIN_REDYELLOWREDDKNKAGREENNKAGREENNKAGREENNKAGREENNKAGREENNKAGREEN	GREEN TIMER	YELLOW TIMER	RED TIMER
//MEMory-element-names         PRINT_         MIN_GREEN_TIME       MAX_GREEN_TIME         MIN_TELLOW_TIME       MAX_YELLOW_TIME_         MIN_TELLOW_TIME       MAX_YELLOW_TIME_         MIN_RED_TIME       MAX_YELLOW_TIME_         MIN_RED_TIME       MAX_YELLOW_TIME_         MIN_RETERING_TIME       MAX_METERING_TIME_         CUR_METERING_TIME       MAX_METERING_TIME_         MEASURE_PERIOD       PERIOD         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_OLD_       FLOW_CUR_         SET_ON_FLOW_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         SPEED_OLD_       VEH_SPEED_OLD_       SPEED_CUR_         SPEED_UPSTREAM_       SPEED_DOUNSTREAM_       SPEED_CUR_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         SET_ON_SPEED_       SET_AIDECREASE_FLOW_       BETA_DECREASE_SPEED_         BETA_INCREASE_SPEED_       BETA_       DECREASE_SPEED_         BETA_INCREASE_SPEED_       BETA_       DCCUPANCY_SETPOINT_         ALPHA_INCREASE_SPEED_       OCCUPANCY_SETPOINT_       ALINEA_CONSTANT_         ONRAMP_VOLUME_       MAX_GREEN_	METERING TIMER	MEASURE TIMER	~~~~ <u>~</u>
<pre>//MEMory-element-names PRINT_ MIN_GREEN_TIME_ MAX_GREEN_TIME_ MIN_WED_TIME_ MAX_YELLOW_TIME_ MIN_WEDTERING_TIME_ MAX_YELLOW_TIME_ MIN_WETERING_TIME_ MAX_METERING_TIME_ CUR_METERING_TIME_ METERING_TIME_ CUR_METERING_TIME_ METERING_TIME_ ELOW_METERING_TIME_ CLEARANCE_TIME_ FLOW_OER_HOUR_ SET_ON_FLOW_ FLOW_OLD_ FLOW_CUR_ SET_ON_FLOW_ SET_OFF_FLOW_ SPEED_NEW_ SPEED_OLD_ SPEED_CUR_ SPEED_NEW_ SPEED_OLD_ SPEED_CUR_ SPEED_NEW_ SPEED_OLD_ SPEED_CUR_ SET_ON_SPEED_ SET_OFF_SPEED_ SET_ON_SPEED_ SET_OFF_SPEED_ SET_ON_SPEED_ SET_OFF_SPEED_ SET_ON_SPEED_ SET_OFF_SPEED_ ALPHA_INCREASE_FLOW_ ALPHA_DECREASE_FLOW_ BETA_INCREASE_SPEED_ BETA_ CAPACITY_ OCCUPANCY_CUR_ OCCUPANCY_SETPOINT_ ALINEA_CONSTANT_ ONRAMP_VOLUME_ //LOGical-element-names _GREENYELLOW_ RED_ DV _ DP_ DA_ DK_ MIN_GREEN_ MAX_GREEN_ MIN_YELLOW_ VEHLOW_ MIN_RED VEHICAL SETANTED </pre>			
MIN_GREEN_TIME       MAX_GREEN_TIME_         MIN_YELLOW_TIME_       MAX_YELLOW_TIME_         MIN_RED_TIME_       MAX_METERING_TIME_         MIN_RETERING_TIME_       MAX_METERING_TIME_         MEASURE_PERIOD       PERIOD         MEASURE_PERIOD       PERIOD         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         SPEED_UPSTREAM       SPEED_DOWNSTREAM_       SPEED_CUR_         SET_ON_SPEED_       SET_MAX_OFF_SPEED_       SET_OFF_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       SETA_DECREASE_FLOM_         ALPHA_INCREASE_FLOW_       ALPHA DECREASE_FLOM_       BETA_DECREASE_SPEED_         BETA_INCREASE_SPEED_       BETA_DECREASE_SPEED_       BETA_         CAPACITY_OCCUPANCY_CUR_       OCCUPANCY_SETPOINT_       ALINEA_CONSTANT_         OWRAMP_VOLUME_	//MEMory-element-name PRINT	S	
MIN_YELLOW_TIME_       MAX_YELLOW_TIME_         MIN_RED_TIME_       MAX_METERING_TIME_         MIN_METERING_TIME_       METERING_TIME_         MEASURE_PERIOD_       PERIOD_         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_         SET_ON_FLOW_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         VEH_SPEED_OLD_       VEH_SPEED_CUR_       SPEED_CUR_         SPEED_UPSTREAM_       SPEED_DOWNSTREAM_       SET_OFF_SPEED         SET_ON_SPEED_       SET_OFF_SPEED       SET_ON_SPEED         SET_ON_SPEED_       SET_OFF_SPEED       SET_OFF_SPEED         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_SPEED       BETA_DECREASE_SPEED         ALPHA_INCREASE_SPEED_       BETA_DECREASE_SPEED       BETA         CAPACITY	MIN GREEN TIME	MAX GREEN TIME	
MIN_RED_TIME_       MAX_METERING_TIME_         MIN_METERING_TIME_       MAX_METERING_TIME_         CUR_METERING_TIME_       METERING_TIME_         MEASURE_PERIOD_       PERIOD_         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_         SPEED_NEW_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         VEH_SPEED_OLD_       VEH_SPEED_CUR_       SPEED_CUR_         SPEED_UPSTREAM_       SPEED_DOWNSTREAM_       SPEED_SET_ON_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_FLOW_       BETA_DECREASE_SPEED_         BETA_INCREASE_SPEED_       BETA_DECREASE_SPEED_       BETA_         CAPACITY_       OCCUPANCY_SETPOINT_       ALPHA_         OCCUPANCY_CUR_       OCCUPANCY_SETPOINT_       DK_         MIN_GREENYELLOWREDDK_      DK_        DPDADKNAX_YELLOWMAX_YELLOWMAX_YELLOWMAX_YELLOWNAX_YELLOWNAX_YELLOWDK_	MIN YELLOW TIME	MAX YELLOW TIME	
MIN_METERING_TIME_       MAX_METERING_TIME_         CUR_METERING_TIME_       METERING_TIME_         MEASURE_PERIOD       PERIOD         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_         SET_ON_FLOW_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         VEH_SPEED_OLD_       VEH SPEED_CUR_       SPEED_CUR_         SET_MAX_ON_SPEED_       SET_MAX_OFF_SPEED_       SET_ON_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_FLOW_       BETA_DECREASE_SPEED_         BETA_INCREASE_SPEED_       BETA_DECREASE_SPEED_       BETA_         CAPACITY_       OCCUPANCY_SETPOINT_       ALPHA_         OCCUPANCY_CUR_       OCCUPANCY_SETPOINT_         ALINEA_CONSTANT_       ONRAMP_VOLUME_         //LOGical-element-names	MIN RED TIME		
CUR_METERING_TIME_       METERING_TIME_         MEASURE_PERIOD       PERIOD         MEASURE_PERIOD       PERIOD         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         VEM_SPEED_OLD_       VEH_SPEED_CUR_       SPEED_CUR_         SPEED_UPSTREAM_       SPEED_DOWNSTREAM_       SPEED_DOWNSTREAM_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_SPEED_       BETA_DECREASE_SPEED_         ALPHA_       OCCUPANCY_CUR_       OCCUPANCY_SETPOINT_         ALPHA_       OCCUPANCY_CUR_       OCCUPANCY_SETPOINT_         ALPHA_       OPPDADK_	MIN METERING TIME	MAX_METERING_TIME_	
MEASURE_PERIOD       PERIOD       PERIOD         MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW       FLOW_OLD       FLOW_CUR         FLOW_PER_HOUR       SET_OFF_FLOW       SPEED_OLD         SPEED_NEW       SET_OFF_FLOW       SPEED_CUR         SPEED_NEW       SPEED_OLD       SPEED_CUR         VEM_SPEED_OLD       SPEED_OUPSTREAM       SPEED_DOWNSTREAM         SET_MAX_ON_SPEED       SET_OFF_SPEED       SET_OFF_SPEED         SET_ON_SPEED       SET_OFF_SPEED       SET_OFF_SPEED	CUR_METERING_TIME_	METERING_TIME_	
MEASURE_MET_TIME_       CLEARANCE_TIME_         FLOW_NEW       FLOW_OLD       FLOW_CUR         FLOW_PER_HOUR       SET_OFF_FLOW       SPEED_OLD       SPEED_CUR         SPEED_NEW       SPEED_OLD       SPEED_CUR       SPEED_CUR         VEH_SPEED_OLD       VEH_SPEED_CUR       SPEED_CUR       SPEED_CUR         SPEED_UPSTREAM       SPEED_DOWNSTREAM       SPEED_CUR       SPEED_CUR         SET_ON_SPEED       SET_MAX_OFF_SPEED       SET_OFF_SPEED       SET_ON_SPEED         ALPHA_INCREASE_FLOW       ALPHA_DECREASE_FLOW	MEASURE PERIOD	PERIOD	
FLOW_NEW_       FLOW_OLD_       FLOW_CUR_         FLOW_PER_HOUR_       SET_OFF_FLOW_       SPEED_OLD_       SPEED_CUR_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_       SPEED_CUR_         VEH_SPEED_OLD_       VEH_SPEED_CUR_       SPEED_CUR_         SET_MAX_ON_SPEED_       SET_MAX_OFF_SPEED_       SET_ON_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_FLOW_       BETA_DECREASE_SPEED_         BETA_INCREASE_SPEED_       BETA_DECREASE_SPEED_       BETA_         CAPACITY_       OCCUPANCY_SETPOINT_       ALINEA_CONSTANT_         ONRAMP_VOLUME_       //LOGical-element-names	MEASURE_MET_TIME_	CLEARANCE_TIME_	
FLOW_NEWFLOW_OLDFLOW_CUR			
FLOW_PER_HOUR_       SET_OFF_FLOW_         SET_ON_FLOW_       SET_OFF_FLOW_         SPEED_NEW_       SPEED_OLD_       SPEED_CUR_         SPEED_OLD_       VEH_SPEED_CUR_       SPEED_OLD_         SPEED_UPSTREAM_       SPEED_DOWNSTREAM_       SPEED_SET_MAX_OFF_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_       SET_ON_SPEED_         SET_ON_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_FLOM_         BETA_INCREASE_SPEED_       BETA_DECREASE_SPEED_         ALPHA_       NCREASE_SPEED_       BETA_DECREASE_SPEED_         ALPHA_       OCCUPANCY_SETPOINT_         ALPHA_       OCCUPANCY_SETPOINT_         ALINEA_CONSTANT_       OCCUPANCY_SETPOINT_         ALINEA_CONSTANT_       OVCUPANCY_SETPOINT_         VLOGical-element-names	FLOW_NEW_	FLOW_OLD_	FLOW CUR
SET_ON_FLOW	FLON_PER_HOUR_		
SPEED_NEW	SET_ON_FLOW_	SET_OFF_FLOW_	
SPEED_NEW			
VEM_SPEED_OLDVEM_SPEED_CUR SPEED_UPSTREAMSPEED_DOWNSTREAM SET_MAX_ON_SPEEDSET_MAX_OFF_SPEED SET_ON_SPEEDSET_OFF_SPEED ALPHA_INCREASE_FLOWALPHA_DECREASE_FLOW BETA_INCREASE_SPEEDBETA_DECREASE_SPEED ALPHABETA CAPACITYOCCUPANCY_SETPOINT ALINEA_CONSTANTOCCUPANCY_SETPOINT ALINEA_CONSTANTOCCUPANCY_SETPOINT ALINEA_CONSTANTORCUPANCY_SETPOINT ALINEA_CONSTANT ONRAMP_VOLUMEREDDK DVDPDADK MIN_GREENMAX_GREENDKMAX_TELLOWMAX_TELLOWMAX_TELLOWMAX_TELLOWNDK	SPEED_NEW_	SPEED_OLD_	SPEED_CUR_
SPEED_UPSTREAMSPEEDSPEEDSET_MAX_ON_SPEEDSET_MAX_OFF_SPEED         SET_MAX_ON_SPEEDSET_OFF_SPEED         ALPHA_INCREASE_FLOWALPHA_DECREASE_FLOWBETA_INCREASE_SPEEDBETA_DECREASE_SPEED         ALPHABETA_DECREASE_SPEEDBETA         CAPACITYOCCUPANCY_SETPOINT         OCCUPANCY_CUROCCUPANCY_SETPOINT         ALINEA_CONSTANT	VEH_SPEED_OLD_	VEH_SPEED_CUR_	
SET_MAX_ON_SPEED	SPEED_UPSTREAM_	SPEED_DOWNSTREAM_	
SET_ON_SPEED_       SET_OFF_SPEED_         ALPHA_INCREASE_FLOW_       ALPHA_DECREASE_FLOW_         BETA_INCREASE_SPEED_       BETA_DECREASE_SPEED_         ALPHA_       BETA_         CAPACITY_       OCCUPANCY_SETPOINT_         ALINEA_CONSTANT_       ONRAMP_VOLUME_         //LOGical-element-names      REDDADK        DVDPDADK      DKDK        MIN_GREENMAX_GREENMAX_YELLOWMAX_YELLOWMAX_YELLOWNAX_YELLOWNIN_RED      NIN_CONTANTED	SET_MAX_ON_SPEED_	SET_MAX_OFF_SPEED_	
ALPHA_INCREASE_FLOWALPHA_DECREASE_FLOW BETA_INCREASE_SPEEDBETA_DECREASE_SPEED ALPHABETA CAPACITYOCCUPANCY_SETPOINT ALINEA_CONSTANTOCCUPANCY_SETPOINT ALINEA_CONSTANTONRAMP_VOLUME //LOGical-element-names GREENYELLOWRED _DVDPDADK _DVDPDADK _MIN_GREENMAX_GREENMAX_USELLOW MIN_REDVELLOW MIN_REDVELICW	SET_ON_SPEED_	SET_OFF_SPEED_	
ALPHA_INCREASE_FLOW_ ALPHA_DECREASE_FLOW_ BETA_INCREASE_SPEED_ BETA_DECREASE_SPEED_ ALPHA_ BETA_ CAPACITY_ OCCUPANCY_CUR_ OCCUPANCY_SETPOINT_ ALINEA_CONSTANT_ ONRAMP_VOLUME_ //LOGical-element-names _GREENYELLOWRED_ _DVDPDADK_ _MIN_GREENMAX_GREEN_ _MIN_GREENMAX_YELLOW_ MIN_RED_ VEHICLE DEPARTED			
BETA_INCREASE_SPEED_ BETA_DECREASE_SPEED_ ALPHABETA CAPACITYOCCUPANCY_SETPOINT ALINEA_CONSTANTONRAMP_VOLUME ONRAMP_VOLUME //LOGical-element-names GREENYELLOWRED _DVDPDADK _DVDPDADK _MIN_GREENMAX_GREEN _MIN_TELLOWMAX_YELLOW MIN_REDVENICLE DEPARTED	ALPHA_INCREASE_FLOW_	ALPHA_DECREASE_FLOW	-
ALPHA_ BETA_ CAPACITY_ OCCUPANCY_CUR_ OCCUPANCY_SETPOINT_ ALINEA_CONSTANT_ ONRAMP_VOLUME_ //LOGical-element-names _GREENYELLOWRED_ _DVDPDADK_ _MIN_GREENMAX_GREEN_ _MIN_YELLOWMAX_YELLOW_ MIN_REDVENTCLE DEPARTED	BETA_INCREASE_SPEED_	BETA_DECREASE_SPEED	-
CAPACITY_ OCCUPANCY_CUR_OCCUPANCY_SETPOINT_ ALINEA_CONSTANT_ ONRAMP_VOLUME_ //LOGical-element-names _GREENYELLOWRED_ _DVDPDADK_ _MIN_GREENMAX_GREEN_ _MIN_GREENMAX_YELLOW_ _MIN_REDVENTCLE_DEPARTED	ALPHA_	BETA_	
LAPACIIT OCCUPANCY_CUR_ OCCUPANCY_SETPOINT_ ALINEA_CONSTANT_ ONRAMP_VOLUME_ //LOGical-element-names _GREEN_ YELLOW_ RED_ _DVDPDADK_ DVDPDADK_ MIN_GREENMAX_GREEN_ MIN_YELLOW_ MIN_REDVELLOW_ MIN_REDVELLOW_	CADACITY		
OLCUPARCT_CUR_     OLCUPARCT_SETPOINT_       ALINEA_CONSTANT_     ONRAMP_VOLUME_       //LOGical-element-names		OCCUDANCY SETDOINT	
ALINEA_CONSTANT_ ONRAMP_VOLUME_ //LOGical-element-names _GREENYELLOWRED_ _DVDPDADK_ _MIN_GREENMAX_GREEN _MIN_YELLOWMAX_YELLOW MIN_REDVENICLE_DEPARTED		OCCOPANCY_SETPOINT_	
UNKAMP_VOLOWE_ //LOGical-element-names _GREENYELLOWRED _DVDPDADK _MIN_GREENMAX_GREEN _MIN_YELLOWMAX_YELLOW_ _MIN_REDVENICLE_DEPARTED	ALINEA_CONSTANT_		
//LOGical-element-names _GREENYELLOWRED _DVDPDADK _MIN_GREENMAX_GREEN _MIN_YELLOWMAX_YELLOW MIN_REDVENICLE_DEPARTED	UNKAMP_VOLUME_		
_GREENYELLOWRED _DVDPDADK _MIN_GREENMAX_GREEN _MIN_YELLOWMAX_YELLOW MIN_REDVENICLE_DEPARTED	///OGical-alement-nem	96	
	GREEN YELL	OU PEN	
			הג
MIN_YELLOWMAX_YELLOW MIN_YELLOWMAX_YELLOW MIN_REDVENICLE_DEPARTED			
		YELLOU	
NETERING NETER CLEARED			

\_PROC\_\_\_\_DUMMY\_ \_AI.TERNATE\_\_\_\_\_ \_SPEED\_LEVEL\_\_\_\_\_ RWS\_\_\_\_\_ALINEA\_\_\_\_

//COLLECTION-names METERING\_INSTALL\_METERING\_SIGNAL\_

In the MANDAT dataset the names of the elements are chosen in a more or less selfexplanatory way. The meaning of the elements is explained in the next section. e.g. the CONDAT dataset.

//\$\$\$/00 statements on intersection level

S(PERIOD_=3600/MEASURE_PERIOD_)	<pre>.= S(INIT_TIMER_=0)</pre>
S(MEASURE_TIMER_=0)	.= S(INIT_TIMER_=0)
S(CUR_METERING_TIME_=MIN_METERING_TIME_)	.= S(INIT_TIMER_=0)

In this initiation procedure a help element PERIOD\_ is calculated. If for instance the measure period is 60 seconds then PERIOD\_ becomes 60. Further the timer for the measure period is started and the metering time is set to its minimum.

S\_METERING\_,S(METERING\_TIMER\_=0) .= S\_GREEN\_\$\$ E\_METERING\_ .= S(METERING\_TIMER\_=CUR\_METERING\_TIME\_)

In these two statements an element \_METERING\_ is introduced, which makes sure that the next green period of a movement can only start when the timer for the metering time has reached a certain value. The current metering time is calculated in the CONDAT dataset.

//\$\$\$/\$\$ statements on signal level
S\_CLEARED\_,S\_RED\_ .= S(INIT\_TIMER\_=0)

This part of the MANDAT dataset concerns the initiation statements on signal level. It is stated that the intersection is cleared and that all signals are set on red.

S_MIN_GREEN_,S_MAX_GREEN_,S(GREEN_TIMER_=0)	.≠ S_GREEN_
E_MIN_GREEN_	.≠ S(GREEN_TIMER_=MIN_GREEN_TIME_)
E_MAX_GREEN_	.≠ S(GREEN_TIMER_=MAX_GREEN_TIME_)
S_MIN_YELLOW_,S_MAX_YELLOW_,S(YELLOW_TIMER_=0)	) .= S_YELLOW_
E_MIN_YELLOW_	.= S(YELLOW_TIMER_=MIN_YELLOW_TIME_)
E_MAX_YELLOW_	.= S(YELLOW_TIMER_=MAX_YELLOW_TIME_)
S_MIN_RED_,S(RED_TIMER_=0) .:	= S_RED_
E_MIN_RED:	= S(RED_TIMER_=MIN_RED_TIME_)

//END

In the last part of the MANDAT dataset the logical elements \_MIN\_GREEN\_ and \_MAX\_GREEN\_ are set to TRUE and the GREEN\_TIMER\_ is activated when the logical element \_GREEN\_ of a signal movement is set to TRUE. This means that if \_MIN\_GREEN\_ is TRUE the timer is running. The elements \_MIN\_GREEN\_ and \_MAX\_GREEN\_ are set to

FALSE when the GREEN\_TIMER\_ reaches a preset time, the minimum green time and maximum green time respectively. The same is applied to the yellow and red periods of the signal movement.

### 3.3. CONDAT dataset

#### 3.3.1. Initiation part

//CONDAT-dataset for a PROTOTYPE of an isolated RAMP-METERING INSTALLATION, with the RWS and ALINEA control strategy. //000 /METERING\_INSTALL\_: 001 //001 /METERING\_SIGNAL\_00: 01 05 45

The first record is on network level and states that there is one location with a metering installation. The second record states that there are three signals on this site (or intersection) 001, namely 01, 05 and 45, where 01 is a dummy signal, 05 is the signal for the metering light and 45 for the bus light.

/RWS\_ = T /ALINEA\_ = F

Here a choice can be made between the Rijkswaterstaat and ALINEA algorithm. If RWS\_ is TRUE the Rijkswaterstaat strategy and if ALINEA is TRUE the ALINEA strategy is in use.

/CAPACITY	= 4400
ALINEA CONSTANT	= 70
/OCCUPANCY SETPOINT	= 0.18
/ONRAMP VOLUME	= 800
/MEASURE MET TIME	= 1.0
/MIN METERING TIME	= 4.5
/MAX METERING TIME	= 12.0
/MEASURE_PERIOD_	= 30.0
/ALPHA_INCREASE FLOW	= 0.25
/ALPHA DECREASE FLOW	= 0.25
/BETA INCREASE SPEED	= 0.2
/BETA DECREASE SPEED	= 0.1

In this part of the CONDAT dataset the intersection memory elements are initiated. The meaning of these elements is:

CAPACITY_	: (fixed) capacity in veh/hour of the motorway used in the
	Rijkswaterstaat control law (see also Deliverable 7a, 3.2.)
ALINEA_CONSTANT_	: positive factor for the integral part of the ALINEA control law
	(see also Deliverable 7a, 3.1.)
OCCUPANCY_SETPOINT_	: set-value for the downstream occupancy (see also
	Deliverable 7a, 3.1.)

MEASURE_MET_TIME_	: step for smoothing the metering time if the calculated metering time is different from the current metering time
MIN_METERING_TIME_ MAX_METERING_TIME_ MEASURE_PERIOD_	<ul> <li>: minimum metering time</li> <li>: maximum metering time</li> <li>: period in which the flow, speed and occupancy are measured and the speed is smoothed; after this period the flow is smoothed and the metering time is calculated</li> </ul>
ALPHA INCREASE FLOW	: factor to smooth an increasing flow
ALPHA DECREASE FLOW	: factor to smooth an decreasing flow
BETA INCREASE SPEED	: factor to smooth an increasing speed
BETA_DECREASE_SPEED_	: factor to smooth an decreasing speed
/SET_ON_FLOW_ = 3500	

/SET	OFF_FLOW_	= 3000
/SET	ON SPEED	= 50.0
/SET	OFF_SPEED_	= 70.0
/SET	MAX_ON_SPEED	= 35
/SET	MAX_OFF_SPEED_	= 50

These values are threshold values for putting the program of the ramp-metering installation in operation and out of operation and for setting the metering time to its maximum value and back to its calculated value.

= 2.0
= 2.0
= 1.0
= 1.0
= 5.0
= 5.0
= 0.5
= 0.5
= 2.0
= 2.0
= 2.0
= 2.0

In the above part several memory elements on signal level are initiated. The meaning of these elements is:

MIN_RED_TIME: minimum red time	CLEARANCE_TIME_ MIN_GREEN_TIME_ MAX_GREEN_TIME_ MIN_YELLOW_TIME_ MAX_YELLOW_TIME_ MIN_RED_TIME_	<ul> <li>time needed to clear the intersection</li> <li>minimum green time</li> <li>maximum green time</li> <li>minimum yellow time</li> <li>maximum yellow time</li> <li>minimum red time</li> </ul>
--------------------------------	--	---

DV\_011 = (SPEED\_DETECTOR\_011>0) \_DV\_012 = (SPEED\_DETECTOR\_012>0) \_DV\_013 = (SPEED\_DETECTOR\_013>0) \_DV\_014 = (SPEED\_DETECTOR\_014>0) \_DV\_015 = (SPEED\_DETECTOR\_015>0) \_DV\_016 = (SPEED\_DETECTOR\_016>0) \_DK\_051 = (SHORT\_LOOP\_051>0) \_DK\_053 = (SHORT\_LOOP\_053>0)

_DP_	054	₹	(LONG_	L00P_(	054>0)
_DK_	055	Ξ	(SHORT	_L00P	_055>0)
_DK_	451	×	(SHORT	LOOP	451>0)
"DK	452	∓	(SHORT	LOOP	452>0)
DK	453	Ξ	(SHORT	LOOP	453>0)

In the above statements the detector elements are joined with logical elements to improve readability and to avoid ambiguous warning messages.

The speed detectors are situated on both lanes of the main road 1400 meters and 400 meters upstream the on-ramp and 400 meters downstream the on-ramp.

In reality the occupancy detectors are placed 400 meters downstream the on-ramp and they measure the **time** that they are occupied. Occupancy detectors in FLEXSYT measure the **number of vehicles** that occupy a detector and divide that number through the maximum number of vehicles that can occupy the detector. So in the program the occupancy detectors are placed right downstream the on-ramp and are 600 meters long. This length guarantees an accuracy of one percent, because the vehicles in FLEXSYT are six meters long.

The short loop detectors and the long loop detector on the on-ramp are used for counting the arrivals, for an early detection, for a demand, for starting yellow, for starting red and for counting the departures.

The situation of all the detectors is shown in figure 3



Figure 3. Situation of detectors in the simulation

#### 3.3.2. Counting and measuring part

```
S(FLOW_NEW_016=FLOW_NEW_016+1) .= S_DV_011+S_DV_012
S(FLOW_NEW_014=FLOW_NEW_014+1) .= S_DV_013+S_DV_014
S(FLOW_NEW_052=FLOW_NEW_052+1) .= S_DK_055 + S_DK_453
S(FLOW_NEW_051=FLOW_NEW_051+1) .= S_DK_052 + S_DK_451
```

These statements count the vehicles 1400 and 400 meters upstream the on-ramp and the arriving and departing vehicles on the on-ramp.

page -17-

page -18-

S(VEH\_SPEED\_CUR\_013=SPEED\_DETECTOR\_013) .= S\_DV\_013 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_013.(VEH\_SPEED\_CUR\_013>VEH\_SPEED\_OLD\_013) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_013.(VEH\_SPEED\_CUR\_013<VEH\_SPEED\_OLD\_013) .= S\_DV\_013 S(VEH\_SPEED\_OLD\_013=BETA\_\*VEH\_SPEED\_CUR\_013+(1-BETA\_)\*VEH\_SPEED\_OLD\_013) .= S\_DV\_013 S(VEH\_SPEED\_CUR\_014=SPEED\_DETECTOR\_014) .= S\_DV\_014 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_014.(VEH\_SPEED\_CUR\_014>VEH\_SPEED\_OLD\_014) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_014.(VEH\_SPEED\_CUR\_014<VEH\_SPEED\_OLD\_014) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_014.(VEH\_SPEED\_CUR\_014<VEH\_SPEED\_OLD\_014) .= S\_DV\_014 S(VEH\_SPEED\_OLD\_014=BETA\_\*VEH\_SPEED\_CUR\_014+(1-BETA\_)\*VEH\_SPEED\_OLD\_014) .= S\_DV\_014 S(VEH\_SPEED\_CUR\_015=SPEED\_DETECTOR\_015) .= S\_DV\_015 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_015.(VEH\_SPEED\_CUR\_015>VEH\_SPEED\_OLD\_015) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_015.(VEH\_SPEED\_CUR\_015<VEH\_SPEED\_OLD\_015) S(VEH\_SPEED\_OLD\_015=BETA\_\*VEH\_SPEED\_CUR\_015+(1-BETA\_)\*VEH\_SPEED\_OLD\_015) .= S\_DV\_015 S(VEH\_SPEED\_OLD\_015=BETA\_\*VEH\_SPEED\_CUR\_015+(1-BETA\_)\*VEH\_SPEED\_OLD\_015) .= S\_DV\_015 S(VEH\_SPEED\_OLD\_016=SPEED\_DETECTOR\_016) .= S\_DV\_016 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_016.(VEH\_SPEED\_CUR\_016>VEH\_SPEED\_OLD\_015) S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_016.(VEH\_SPEED\_CUR\_016>VEH\_SPEED\_OLD\_016) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_016.(VEH\_SPEED\_CUR\_016<VEH\_SPEED\_OLD\_016))</pre>

S(VEH\_SPEED\_OLD\_016=BETA\_\*VEH\_SPEED\_CUR\_016+(1-BETA\_)\*VEH\_SPEED\_OLD\_016) .= S\_DV\_016

In this part of the CONDAT dataset the individual speed of the vehicles 400 meters upstream and 400 meters downstream the on-ramp is smoothed.

S\_PROC\_,E\_PROC\_,S(MEASURE\_TIMER\_=0) .= S(MEASURE\_TIMER\_=MEASURE\_PERIOD\_)

This statement starts a procedure for smoothing the flow and calculating the metering time after a measure period has passed. After this is done, the measure timer is again started.

S(ALPHA\_=ALPHA\_INCREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_016>FLOW\_OLD\_016) S(ALPHA\_=ALPHA\_DECREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_016<FLOW\_OLD\_016) S(FLOW\_OLD\_016=ALPHA\_\*FLOW\_NEW\_016+(1-ALPHA\_)\*FLOW\_OLD\_016) .= S\_PROC\_ S(FLOW\_CUR\_016=FLOW\_NEW\_016),S(FLOW\_NEW\_016=0) .= S\_PROC\_ S(ALPHA\_=ALPHA\_INCREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_014>FLOW\_OLD\_014) S(ALPHA\_=ALPHA\_DECREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_014>FLOW\_OLD\_014) S(FLOW\_0LD\_014=ALPHA\_\*FLOW\_NEW\_014+(1-ALPHA\_)\*FLOW\_0LD\_014) .= S\_PROC\_ S(FLOW\_CUR\_014=FLOW\_NEW\_014),S(FLOW\_NEW\_014=0) .= S\_PROC\_ S(FLOW\_CUR\_052=FLOW\_NEW\_052),S(FLOW\_NEW\_052=0) .= S\_PROC\_ S(FLOW\_CUR\_051=FLOW\_NEW\_051),S(FLOW\_NEW\_051=0) .= S\_PROC\_

These statements smooth the flow 1400 meters and 400 meters upstream the on-ramp. Before the smoothing is carried out, a smoothing factor is determined. Further the flow on the on-ramp is measured.

S(SPEED\_UPSTREAM\_=(VEH\_SPEED\_OLD\_013+VEH\_SPEED\_OLD\_014)/2) .= S\_PROC\_ S(SPEED\_DOWNSTREAM\_=(VEH\_SPEED\_OLD\_015+VEH\_SPEED\_OLD\_016)/2) .= S\_PROC\_

The average speed upstream and downstream the on-ramp is calculated.

```
S(FLOW_PER_HOUR_ = FLOW_OLD_014 * PERIOD_) .= S_PROC_
S_METER_ .= S_PROC_.((SPEED_UPSTREAM_ < SET_ON_SPEED_) + (SPEED_DOWNSTREAM_ < SET_ON_SPEED_) +
+ (FLOW_PER_HOUR_ > SET_ON_FLOW_))
E_METER_ .= S_PROC_.(SPEED_UPSTREAM_ > SET_OFF_SPEED_) . (SPEED_DOWNSTREAM_ > SET_OFF_SPEED_) .
. (FLOW_PER_HOUR_ < SET_OFF_FLOW_) . _DP_054N</pre>
```

In this procedure it is determined whether or not ramp-metering is necessary. Therefor the flow per hour is calculated with the help of the element PERIOD\_. The installation program is activated when a measure period is over and if the (smoothed) upstream speed is below a preset value or if the (smoothed downstream speed is below the same value or if the (smoothed) upstream flow is above a preset value. The ramp-metering installation program is deactivated when a measure period is over and if the upstream speed is above a certain value and if the downstream speed is above the same value and if the upstream flow goes is a preset value. But this is only done if there is no queue before the stopline on the on-ramp.

S\_SPEED\_LEVEL\_ .= S\_PROC\_.((SPEED\_UPSTREAM\_SET\_MAX\_ON\_SPEED\_)+(SPEED\_DOWNSTREAM\_SET\_MAX\_ON\_SPEED\_))
E\_SPEED\_LEVEL\_ .= S\_PROC\_.(SPEED\_UPSTREAM\_SET\_MAX\_OFF\_SPEED\_).(SPEED\_DOWNSTREAM\_SET\_MAX\_OFF\_SPEED\_)

After every measure period it is determined whether or not the metering time must be set to its maximum value. This is the case if the upstream or the downstream speed is below a certain value. The metering time is reset to its calculated value if both the upstream and downstream speed are above a preset value.

```
S(ONRAMP_VOLUME_=CAPACITY_-FLOW_PER_HOUR_)
```

.= S\_PROC\_.RWS\_.\_SPEED\_LEVEL\_N

In the Rijkswaterstaat algorithm the allowed on-ramp volume is calculated directly from the capacity and the smoothed upstream flow (see Deliverable 7a, 3.2.).

S(OCCUPANCY\_CUR\_=(OCCUPANCY\_DETECTOR\_011+OCCUPANCY\_DETECTOR\_012)/2) .= S\_PROC\_.ALINEA\_ S(ONRAMP\_VOLUME\_=ONRAMP\_VOLUME\_+ALINEA\_CONSTANT\_\*(OCCUPANCY\_SETPOINT\_-OCCUPANCY\_CUR\_)) .= S\_PROC\_.ALINEA\_

In the ALINEA algorithm the allowed on-ramp volume is calculated from the occupancy (see Deliverable 7a, 3.1.).

S(METERING_TIME_=3600/ONRAMP_VOLUME_)	.= S_PROC_
S(METERING_TIME_=MIN(METERING_TIME_,MAX_METERING_TIME_))	.= S_PROC_
S(METERING_TIME_=MAX(METERING_TIME_,MIN_METERING_TIME_))	.= S_PROC_
S(METERING_TIME_=MAX_METERING_TIME_)	.= S_PROCRWSSPEED_LEVEL

Then the metering time is calculated and compared with a minimum and maximum value. The metering time is set to its maximum value when <u>SPEED\_LEVEL</u> is TRUE, but only in the Rijkswaterstaat algorithm (see Deliverable 7a, 3.3.).

#### 3.3.3. Signal handling part

```
S_DUMMY_012,S(CUR_METERING_TIME_=CUR_METERING_TIME_+MEASURE_MET_TIME_) .= (E_RED_05+E_RED_45).
.(METERING_TIME_>(CUR_METERING_TIME_+MEASURE_MET_TIME_)).(_DK_053+_DK_452)
S_DUMMY_012,S(CUR_METERING_TIME_=CUR_METERING_TIME_-MEASURE_MET_TIME_) .= (E_RED_05+E_RED_45).
.(METERING_TIME_<(CUR_METERING_TIME_-MEASURE_MET_TIME_)).(_DK_053+_DK_452)
S(CUR_METERING_TIME_=(DERING_TIME_) .= (E_RED_05+E_RED_45).
.DUMMY_012 .= E_RED_05 + E_RED_45
```

Every green period of the metering light or the bus light it is determined whether or not the metering time must be adjusted. Therefor a dummy element is used. The metering time is stepwise adjusted to the calculated metering time, but only when a detector just before a stopline is occupied, otherwise it is done directly.

```
S_GREEN_45
                                                   = E RED 45
                                                   .= S_GREEN 45
E_CLEARED_45
E_GREEN_45 .= (E_MIN_GREEN_45._VEHICLE_DEPARTED_45 + S_DK_451._MAX_GREEN_45 + E_MAX_GREEN_45)._METER_
               + S METER
S_YELLOW_45,E_ALTERNATE_05,E_VEHICLE_DEPARTED_45 .= E_GREEN_45
                                                   .= E_GREEN_45 .
S_ALTERNATE_45
                                                                    _DK_053
                                                   .= E_GREEN_45 . _DO
E YELLOW 45
S RED 45
                                                   .= E YELLOW 45
E RED 45
           .= ((S_DK_452+E_METERING_+E_MIN_RED_45+S_RED_05+S_CLEARED_05)._DK_452._METERING_N.
               ._MIN_RED_45N._RED_05._CLEARED_05._ALTERNATE_45N)._METER_ + E_METER_
S_VEHICLE_DEPARTED_45
                                                   .= S_DK_451 .
                                                                 MIN GREEN 45
                                                   .= S(RED_TIMER_45 = CLEARANCE_TIME 45)
S CLEARED 45
S_GREEN_05
                                                   .= E_RED_05
E_CLEARED_05
                                                    = S_GREEN_05
E_GREEN_05 .= (E_MIN_GREEN_05._VEHICLE_DEPARTED_05 + S_DK_052._MAX_GREEN_05 + E_MAX_GREEN_05)._METER_+
               + S METER
S_YELI.OW_05,E_ALTERNATE_45,E_VEHICLE_DEPARTED_05 .= E_GREEN_05
                                                   .= E_GREEN_05 . _DK_452
.= S_DK_051._MIN_YELLOW_05N + E_MAX_YELLOW_05
S_ALTERNATE_05
E_YELI.OW_05
S_RED_05
                                                    = E YELLOW 05
           .= ((S_DK_053+E_METERING_+E_MIN_RED_05+S_RED_45+S_CLEARED_45)._DK_053._METERING_N.
E_RED_05
               ._MIN_RED_05N._RED_45._CLEARED_45._ALTERNATE_05N)._METER_ + E_METER_
S_VEHICLE_DEPARTED_05
                                                   .= S DK 052 .
                                                                 _MIN_GREEN_05
                                                   .= S(RED_TIMER_05 = CLEARANCE_TIME_05)
S CLEARED 05
//END
```

. \_...

In the last part of the CONDAT dataset the handling of the metering and bus signal is described.

A green period is started when a red period is terminated. The starting of a green period makes the logical element for the clearing of the intersection FALSE. A green period is terminated under three events: when the minimum green time is reached, if the vehicle is departed, when the detector after the stopline becomes occupied, if the green period is in extended green and when the maximum green time is reached.

The termination of a green period starts the yellow period, gives the turn to the other signal, if there is a demand for that signal, and makes the logical element for the departure of a vehicle FALSE. An yellow period is terminated for the bus signal when the minimum yellow time is reached and for the metering signal when the detector six meters after the stopline is occupied, if the minimum yellow time is reached and when the maximum yellow time is reached.

The termination of an yellow period starts a red period. A red period is terminated when

DRIVE-project V1035 "CHRISTIANE", to be included in Deliverable 7k	)
A software prototype for isolated ramp-metering	

the detector for the stopline is occupied, when the metering time is reached, when the minimum red time is reached, when the red phase of the other signal is started or when the intersection is cleared. But also several conditions must be satisfied: the detector before the stopline must be occupied, the metering time and the minimum red time must have been reached, the other signal must be red, the intersection must have been cleared and the signal must have the turn. The red period is also ended when ramp-metering is no longer necessary.

A vehicle is said to be departed when the detector after the stopline becomes occupied within the minimum green time.

The intersection is said to be cleared when the red timer of the signal has reached a certain preset value.

### 3.4. Conclusion

This completes the software prototype for an isolated ramp-metering installation. With this deliverable a floppy diskette is included. On this diskette a demonstration of the prototype can be seen on the Coentunnel sight and further information on FLEXSYT, input and output data sets is available. Just put the diskette in a station and type **go**<**RETURN**> after the DOS-prompt. A screen will be shown which gives information about how the demonstration can be started.

If the demonstration is running several things can be seen. When a detector changes colour (from grey to white) a vehicle is detected. When there is no ramp-metering the lights are both green. Whether or not there is ramp-metering can be seen in the right lower corner of the screen. When there is ramp-metering the light before the strategy that is used is green and the others are red. When there is no ramp-metering the light before **no ramp-metering** is green.

The state of demonstration can be changed with function key  $\langle F4 \rangle$  from SPEEDY to REAL TIME and back. With function key  $\langle F3 \rangle$  a situation can be held and with  $\langle F1 \rangle$  help is available. With  $\langle F2 \rangle$  the demonstration can be stopped.

The screen which appears in the demonstration on the floppy is shown in figure 7.



Figure 4. Demonstration of the software prototype

# APPENDICES

# A. Examples

### A.1. MANDAT dataset

```
//MANDAT dataset for FIXED-TIME control
//DET
          DV DP DA DK
     DS
 //TIM
     CYCLE_timer YELLOW_timer
 //MEM
    PRINT CYCLE_time GO_time NOGO_time
 //LOG
     GO_signal
DS_DV_DP_DA_DK_
//COL
     CONTROLLERS SIGNALS
 //$$$/$$
S(PRIN11='*') .= SDS_2
S(PRIN11='=') .= SDP_2+S(PRINT1=' ').DP_2.DS_2N
S(PRIN11='>') .= SDK_2+S(PRINT1=' ').DK_2.DP_2N.DS_2N
S(PRINT1=' ') .= EDS_2+EDP_2+EDK_2
S(PRINT2='#') .= SGO_signal
S(PRINT2='x') .= EGO_signal
S(PRINT2='') .= S(YELLOW_timer=3)
S(PRINT3='*') .= SDS_1
S(PRINT3='=') .= SDP_1+S(PRINT3='').DP_1.DS_1N
S(PRINT3='<') .= SDK_1+S(PRINT3='').DK_1.DP_1N.DS_1N
S(PRINT3='') .= EDS_1+EDP_1+EDK_1
SGO_signal
                       .= S(CYCLE_timer=GO_time)
EGO_signal
                       .= S(CYCLE_timer=NOGO_time)
S(YELLOW_timer=0).= EGO_signal
S(CYCLE_timer=0) .= S(CYCLE_timer=CYCLE_time$$$/00)
```

//END

# A.2. CONDAT dataset

//CONDAT dataset for FIXED-TIME control of the TRANSYT network

//000 /CONTROLLERS: /CYCLE_time <b>\$\$\$/</b> :	001 72	00 7	2 2	003 36
//001 /SIGNALS: /PRINT\$\$= /GO_time\$\$= /NOGO_time\$\$=	02 1 22 41	05 2 47 16	08 3 22 41	11 4 47 19
;detector interfa DS_111 = (DS111>( DS_112 = (DS112>(	ace )) ))			
DP_021 = (DP021>( DP_051 = (DP051>( DP_081 = (DP081>( DP_111 = (DP111>(	)) )) ))			
DK_021 = (DK021>( DK_051 = (DK051>( DK_081 = (DK081>( DK_111 = (DK111>(	)) )) ))			
//002 /SIGNALS: /PRINT\$\$= /GO_time\$\$= /NOGO_time\$\$=	05 6 11 61	08 7 66 5	11 8 11 61	
;detector interfa DS_111 = (DS111>0 DS_112 = (DS112>0	ice )) ))			
DP_051 = (DP051>0 DP_081 = (DP081>0 DP_111 = (DP111>0	) ) )			
DK_051 = (DK051>0 DK_081 = (DK081>0 DK_111 = (DK111>0				
//003 /SIGNALS: /PRINT\$\$= /GO_time\$\$= /NOGO_time\$\$=	02 10 15 31	08 11 15 32	10 12 0 12	12 13 0 11
;detector interfa DP_021 = (DP021>0 DP_081 = (DP081>0 DP_101 = (DP101>0 DP_121 = (DP121>0	ce ) ) )			
DK_021 = (DK021>0 DK_081 = (DK081>0 DK_101 = (DK101>0 DK_121 = (DK121>0	) ) )			

//END

# A.3. NETDAT dataset

//NETDAT dataset of the TRANSYT network (100%)	traffic	load)
--	---------	-------

; VOR 240	1 I M 240	NRN B 1 1	L1 E 00 1	8L2 RI 100	EG RI 1	ES   0 12	RND 345										
//GEN ;GNUM 102 105 108 208 211 251 302	GT - 10 - 10 - 20 - 21 - 21 - 30	OL GT 25 54 84 84 15 15 24	YP > 1 1 1 1 2 1	50 350 300 200 840 60 200	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	GI	NT 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	- < 0 0 0 0 0 0 0		
//NET ;LNUM -1054 1053 1052 1051	LTH 999 300 30 20	LSAT 1800 1800 1800 1200	CON 0 001 001	I SGL 0 0 05 05	>-L) 40 40 40 40	/TM-< 0 0 0 0	>-1	EQL 0 0 0 0	- < 0 0 0	> 0 0 0 0	-LTOL 1053 1052 1051 2053	< 0 0 3083	> 0 0 50	350 350 350 150	-LTFF 0 0 105	!	~
2053 2052 2051	350 30 20	1800 1800 1500	0 002 002	0 05 05	42 42 42	0 0 0		0 0 0	0 0 0	0 0 0	2052 2051 0	0 0 3114	0 0 0	250 250 150	0 0 100		
- 1084 1083 1082 1081	999 300 30 20	1800 1800 1800 1500	0 0 001 001	0 0 08 08	34 34 34 34	0 0 0 0		0 0 0 0	0 0 0 0	0 0 2053	1083 1082 1081 3083	0 0 0 0	0 0 50	300 300 300 200	0 0 0 50		
3083 3082 3081	220 30 20	1800 1800 1800	0 003 003	0 08 08	37 37 37	0 0 0		0 0	0 0	0 0	3082 3081	0 0	0 0	350 350	0 0		
-2084 2083 2082 2081	999 300 30 20	1800 1800 1800 1500	0 0 002 002	0 0 08 08	39 39 39 39	0 0 0 0		0 0 0 0	0 0 0 0	0 0 0 0	2083 2082 2081 3114	0 0 1115	0 0 50	200 200 200 100	0 0 0 50		
3114 3103 3102 3101 3123 3122 3121	250 10 30 20 10 30 20	1800 1800 1800 1200 1800 1800 1500	0 003 003 003 003 003	0 10 10 10 12 12	42 42 42 42 42 42 42	0 0 0 0 0 0		0 0 0 0 0	0 0 0 0 0 0	3103 0 0 3122 3121	3123 3102 3101 0 0 0	0 0 1024 0 0	100 0 0 100 100	250 350 250 0 0 0	0 0 100 0 0		
-2115 2114 2113 2112 2111	999 150 150 30 20	1800 1800 1800 1800 1800 1500	0 002 0 002 002	0 11 0 11 11	39 39 39 39 39 39	36 36 36 36 36		0 0 0 0	0 0 0 0 0	0 0 0 3114	2114 2113 2112 2111 2111 1115	0 0 0 0 0	0 0 0 150	840 840 840 840 690	0 0 0 0	0 60 0 60 0 60 0 60 0 60	0 0 0 0
1115 1114 1514 1113 1112 1111	188 12 12 150 30 20	1800 1800 720 1800 1800 1500	0 001 001 001 001 001	0 11 11 11 11 11	43 43 43 43 43 43	36 0 -20 36 36 36		0 0 0 0	0 0 0 0	0 0 0 0	1114 1113 1113 1112 1112	1514 0 0 0	0 0 0 0	740 740 0 740 740	0 0 0 0	0 0 6 0 0 0 60 0 60 0 60	0 0 0 0
-3024 3023 3022 3021	999 300 30 20	1800 1800 1800 1800	0 0 003 003	0 0 02 02	36 36 36 36	0 0 0		0 0 0 0	0 0 0 0	0 0 0 0	3023 3022 3021 1023	0 0 0 0	0 0 0 0	200 200 200 200 200	0 0 0		
- 1025 1024 1023 1022	999 150 220 30	720 1800 1800 1800	0 0 0 001	0 0 0 02	25 25 41 41	0 0 0 0		0 0 0 0	0 0 0 0	0 0 0 0	1024 1022 1022 1021	0 0 0 0	0 0 0 0	50 50 300 350	0 0 0 0		

1021 20 1500 001 02 41 0 2053 0 300 50 0 0 0 0 //SIG 1021 1051 1081 1111 2051 2081 2111 3021 3081 3101 3121 //DET /DS1 2111 1111 /DS2 2114 1114 1514 /DP1 1022 1052 1082 1112 2052 2082 2112 3022 3082 3102 3122 /DK1 1021 1051 1081 1111 2051 2081 2111 3021 3081 3101 3121

//END



Figure 5. The network of the NETDAT data set

# A.4. REGFIL dataset

FLEXSYTI-9.0.1. \* TRAFFIC-CONTROL-SIMULATION-PACKAGE \* PART FLXREG \* RWS/DVK\_CX \* ROTTERDAM \* 07-10-90 \* 15:09:37 \* TABLE 1 0NETDAT dataset of the TRANSYT network (100% traffic load)

CONDAT dataset for FIXED-TIME control of the TRANSYT network

MANDAT dataset for FIXED-TIME control

TIME	>	INDICES		<
TIME	001 001	002	005	005
IN	02 08	08	08	12
SECONDS	001 00	1 002 002	003 00	3
	05 1	1 05 11	02 1	0
241	#= < #=	- # = #=	# #	
242	#= < #=	< # #<	# #=	<
243	#= < #=	< # <#=	# #=	<
244	#= < #=	< # <#<	# #=	<
245	#< < #<	< # <#=	#= #=	<
240		~ # <#<	#= #	<
241		~ # ~ #=	#= #<	<
240		~ # < #<	X= #	<
249	# - #~	- # <#-		2
250		- # \#-	<u>^_ \</u>	
252	# < #=		- ^	~
253	# = #<	- + ++	_ #	~ < #
254	#< = #=	- # < #<	< #	< #=
255	#< = #<	- # <#=	< #	· #- < #=
256	# = #	= # <#=	< #	< #=
257	# = #<	= # < #<	< #	#
258	x = x=	= # < #=	< #	
259	x ≈ x=	= # < #<	< #	#
260	x = x=	= # <#=	< = #	#=
261	= = :	= # < #<	< = #	 #=
262	=	= # <#=	< = #	#=
263	= <	= # < #<	< #	#=
264	#= < #	- # <#-	< < #	x=
265	#= < #	= # <#=	< < x	x<
266	#= < #	= # < #<	< < x	x
267	#= < #	= # <#=	< < x	<
268	#= < #	= # < #<	#< #<	<
269	#< = #	< # < #=	#< #=	<
270	#< = #	= # < #<	#< #=	<
271	#= = #	= # <#=	# #=	<
272	#= = #	= # <#=	#< #=	<
273	#= < #	< # <#<	# #=	<
274	#< = #	= # <#=	# #<	<
275	#= = #	< # <#<	# #=	<
2/6	#= = #	= # < #=	# #=	<
2//	#= = #	= # < #<	# #=	<
278	#= = #	< x < x	# #<	<
279	#< = #	= X= < X<	# #=	*
280		< X= < X	# #<	=
201	######################################		# #= 	=
202 297		< = < - ∠#∠	* *	~
203		- 、#*、		2
204 285	- = = = = =	、 、 ₩ = < #<	x #`	2
284		- ` <i>\</i> `	~ ^ × ×	ż
287		· · · · · · · · · · · · · · · · · · ·	ŶŶ	ì
288		- • <del>•</del>	^_	ż
289	<pre></pre>		= #	, #<

#### A.5. RESFIL dataset

FLEXSYTI-9.0.1. \* TRAFFIC-CONTROL-SIMULATION-PACKAGE \* PART FLXRES \* RWS/DVK\_CX \* ROTTERDAM \* 30-09-90 \* 14:37:01 \* page 1 NETDAT dataset of the TRANSYT network (100% traffic load) CONDAT dataset for FIXED-TIME control of the TRANSYT network MANDAT dataset for FIXED-TIME control

<b>&gt;</b>	-NON-(	GREEN	TIMES		< >					-PER	CENT	AGES	-0F-	NON -	GREE		fES-1	IN-C	LASS	SES-0	F-10	- SEC	ONDS-	WIDT	H					<
SIGNAL	NUM		IN SE	CONDS		0/	10/	20/	30/	40/	50/	60/	70/	80/	90,	/100/	110,	/120	/130	)/140	/150	/160	/170/	180/	190/	200	/210	/220	/230,	/240
NUMBER	OF	AV	SD	MIN	MAX	10/	20/	30/	40/	50/	60/	70/	80/	90/	100,	/110/	120	/130	/140	)/150	/160	/170	/180/	190/	200/	210	/220	/230	/240	/INF
	CYCL	>			<	>								• • • •																<
001/02	150	53	0	53	53	0	0	0	0	0	100	0	0	0	0	0	0	0	) (	) (	0	0	0	0	0	0	0	0	0	0
001/05	150	31	0	31	31	0	0	0	100	0	0	0	0	0	0	0	0	0	) (	) (	0	0	0	0	0	0	0	0	0	0
																						-								
>			PER	CENT	AGES	-0F-		GREE	ENTIN	IES-1	IN-C	LASS	SES-0	F-10	-SEC	ONDS-	WIDT	H					<							
SIGNAL	NUM		IN SE	CONDS		0/	10/	20/	30/	40/	50/	60/	70/	80/	90,	/100/	110,	/120	/130	0/140	/150	/160	/170/	180/	190/	200	/210	/220	/230	/240
NUMBER	OF	AV	SD	MIN	MAX	10/	20/	307	40/	50/	60/	/0/	807	90/	100	/110/	120,	/130	/140	1/150	/160	/170	/180/	190/	200/	210	/220,	/230	/240	/INF
004 (00	CYCL	><			<	>												••••												<
001/02	149	19	, n	19	19	U	100	0	U	0	0	U O	0	U	0	0	U	U			U U	U	0	U	U	0	0	0	U	Ŭ
001705	150	41	U	41	41	U	U	U	U	100	U	U	U	U	U	U	U	U	, i	, U	0	U	U	U	U	U	U	U	U	U
<b></b>			TIMES.		/	·				DE	DCEN	TACE	e-0E		cvci	E 71	MEC.	- 1 N -	-		05-1	0-95	CONOS		ти					
STONAL	NEM		TN C	CUNDS		0/	10/	20/	30/		507	607	3-0r 70/	80/		100	110	/120	1130	)/1/0	/150	1160	11701	180/	100	200	/210	/220	/230	/240
NIMRED	OF	A.V	11 31	MIN	MAY	10/	20/	30/	40/	50/	- <u>60</u> /	70/	807	00,	100	1100	120	/130	140	1/150	/160	/170	/180/	100/	200/	210	/220	/230	/240	/185
NONDER	CYCL	><			<	>																								<
001/02	149	72	0	72	72	0	0	0	0	0	0	0	100	0	0	0	0	0	) (	) (	0	0	0	0	0	0	0	0	0	0
001/05	150	72	Ō	72	72	Ō	Ō	Ō	Ó	Õ	Ő	Q	100	Q	Ō	Ō	Ō	Ō	Ċ	) d	0	Ó	0	Ō	Ő	Ó	Ō	Ó	Ō	Ō
FLEXSYT	1-9.0	.1.	* TR/	AFFIC-	CONTR	lol - S	IMUL/	ATIO	N-PA	CKAG	E *	PART	FLX	RES	* RI	IS/D	/K_C)	к *	ROTI	ERDA	M *	30	-09-9	* 0	14	:37	:01	* P	age	2
																	-													
NETDAT	datase	et of	the 1	RANSY	T net	work	(100	0% t	raff	ic l	oad)																			
CONDAT dataset for FIXED-TIME control of the TRANSYT network																														
MANDAT	datase	et fo	r FIX	D-TIM	E cor	ntrol																								
>>>> AT	TENTIO	)N >>	<b>&gt;&gt;&gt;&gt;</b> >	⊳> tra	ffic-	mode	ls of	f FL)	(SYT	- I -	not	scie	ntif	ic v	alio	lated	l, so	o be	car	eful	wit	h re	sults	: <<<	<<<	<<<	ATTE	NTIO	N <<	<<
					_		_	_																						
Results	obta	ined	after	1080	0. se	cond	s of	sim	ulat	ion																				
>	[	DELAY	-AT-SI	OPLIN	E			< >		••••		-PER	CENT	AGES	-OF	DEL	(Y-1)	N-CL	ASSE	S-OF	-5-0	R-10	-SECC	NDS-	WIDT	H				-<
FROM	1	TO T	NUM	_	IN SE	COND	S		0/	5/	10/	15/	20/	30/	40/	50/	60/	70/	80/	90/	100/	110/	120/1	30/1	40/1	50/	160/	170/	180/	190
GENER.	SIGN/	AL Y	OF		V	SD	M	NX .	5/	10/	15/	207	507	40/	<b>50/</b>	60/	/U/	80/	90/	100/	110/	120/	130/1	40/1	50/1	60/	1/0/	180/	190/	INF
>>	001 //	•< P	CARS	>				·< >						• • • •				· · · · ·												~~<
102	001/0	12 1	152	OU.	3 3 F	99.9	188.	.2	5	2	4	2 40	0	10	11	13	ö	2	2	0	2	3	1	4	2	1	U	U	2	Ű
105	001/0	<i>ו</i> כו	1034	11.	ר כ	1.3	- 40.		+2	10	11	10	<b>2</b> U	D	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

1051 001/05

2052 002/05

3 16 37 30 66

5 140 6 2 1

0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Ω 0 0 0 0 0 0

0 0

0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0/ 5/ 10/ 15/ 20/ 30/ 40/ 50/ 60/ 70/ 80/ 90/100/110/120/130/140/150/160/170/180/190 FROM TO T NUM IN SECONDS SIGNAL SIGNAL Y OF AV SD MAX 5/ 10/ 15/ 20/ 30/ 40/ 50/ 60/ 70/ 80/ 90/100/110/120/130/140/150/160/170/180/190/INF -----001/02 002/05 1 139 14.5 n 3.2 15.5 80 17 3 0 0 0 0 0 0 0 0 0 0 0 0 001/05 002/05 1 516 2.1 0 0 0 0 0 0 0 FLEXSYTI-9.0.1. \* TRAFFIC-CONTROL-SIMULATION-PACKAGE \* PART FLXRES \* RWS/DVK CX \* ROTTERDAM \* 30-09-90 \* 14:37:01 \* page 3 NETDAT dataset of the TRANSYT network (100% traffic Load) CONDAT dataset for FIXED-TIME control of the TRANSYT network MANDAT dataset for FIXED-TIME control >>>> ATTENTION >>>>>>>> traffic-models of FLXSYI-I- not scientific validated, so be careful with results <<<<< ATTENTION <<<< Results obtained after 12 subrun(s) of 900, seconds each LANE >----TYPE-TWO------ CDEG SIGNAL AT SAT >--AVERAGE-TOTALS-AT-STOPLINE---< >--AVERAGE-TOTALS-AT-STOPLINE---< OF NUMBER LINK FLOW FLOW >-----DELAY-IN-SECONDS----< FLOW >-----DELAY-IN-SECONDS----< SAT NUM VH/H VH/H NUM AV SD MIN MAX VH/H NUM AV SD MIN MAX 🏅 SUB >----< SUB >-----001/02 1021 1500 336 0 \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* 85 12 65.0 30.2 37.7 129.1 0 1051 1200 345 12 11.4 2.1 8.6 14.5 001/05 0 \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* 50 0 FLEXSYTI-9.0.1. \* TRAFFIC-CONTROL-SIMULATION-PACKAGE \* PART FLXRES \* RWS/DVK\_CX \* ROTTERDAM \* 30-09-90 \* 14:37:01 \* page 3 NETDAT dataset of the TRANSYT network (100% traffic load) CONDAT dataset for FIXED-TIME control of the TRANSYT network MANDAT dataset for FIXED-TIME control >>>> ATTENTION >>>>>>>> traffic-models of FLXSYT-I- not scientific validated, so be careful with results <<<<< ATTENTION <<<< Results obtained after 10800. seconds of simulation LINK SIGNAL MAX 0/ 1/ 2/ 3/ 4/ 5/ 6/ 8/ 10/ 12/ 14/ 16/ 18/ 20/ 22/ 24/ 26/ 31/ 36/ 41/ 46/ 51/ 56/ 61/ 66/ 71/ 76/ NUMBER NUMBER QUEUE 0/ 1/ 2/ 3/ 4/ 5/ 7/ 9/ 11/ 13/ 15/ 17/ 19/ 21/ 23/ 25/ 30/ 35/ 40/ 45/ 50/ 55/ 60/ 65/ 70/ 75/INF/ 1053 50 146 2 1 0 0 0 0 0 0 0 0 0 0 0 0 Ω Ω Ω Ω Λ n Ω Λ Ω 0 0 Ω 1052 001/05 5 109 12 11 10 2 5 0

FLEXSYTI-9.0.1. \* TRAFFIC-CONTROL-SIMULATION-PACKAGE \* PART FLXRES \* RWS/DVK\_CX \* RUTTERDAM \* 30-09-90 \* 14:37:01 \* page 4 NETDAT dataset of the TRANSYT network (100% traffic load) CONDAT dataset for FIXED-TIME control of the TRANSYT network MANDAT dataset for FIXED-TIME control >>>> ATTENTION >>>>>>>> traffic-models of FLXSYI-I- not scientific validated, so be careful with results <<<<<< ATTENTION <<<< Results obtained after 12 subrun(s) of 900. seconds each This table concerns car-type=1 NUM LINK FLOW SAT LINK DIST. DELAY TIME YNEY STOPS MAX QUEUE ALL TYPES OF NUMBER VEH/H FLOW lgth TRVLLD VEH.H/H TIME SPENT VEH/H VEH SUB AV SD MIN MAX VH/H MTRS VH.KM/H AV SD MIN MAX SECS VEH.H/H AV SD MIN MAX % CAP AV SD MIN MAX >--------< >--------->----< 12 1053 346 33 284 408 1800 300 103.80 .03 .01 .02 .06 27.0 2.63 3 12 0 50 1 0 0 0 0 2 1052 345 12 32 284 408 1800 30 .25 10.35 .12 .04 .44 2.7 .51 38 72 11 5 21 0 3 1 0 5 12 1051 344 34 280 408 1200 20 6.89 .79 .14 .60 .99 1.8 .96 104 14 84 132 30 3 3 0 3 3 12 2053 269 32 208 316 1800 350 94.38 .03 .01 .04 30.0 .01 2.28 0 0 0 0 0 58 0 0 0 0 NUM TOTAL TOTAL TOTAL TOTAL SPEED OF DISTANCE TIME DELAY STOPS SUB TRAVELLED SPENT VEH.KM/H VEH.H/H VEH.H/H VEH/H KM/H 12 1337.52 62.69 28.80 2388. 21.51 AVERAGE 65.03 7.12 5.99 363. 1.86 STANDARD DEVIATION 1250.29 51.02 1884. 18.93 MINIMUM 18.96 1469.91 77.64 40.35 3232. 24.74 MAXIMUM FLEXSYTI-9.0.1. \* TRAFFIC-CONTROL-SIMULATION-PACKAGE \* PART FLXRES \* RWS/DVK\_CX \* ROTTERDAM \* 30-09-90 \* 14:37:01 \* page 6 NETDAT dataset of the TRANSYT network (100% traffic load) CONDAT dataset for FIXED-TIME control of the TRANSYT network MANDAT dataset for FIXED-TIME control Results obtained after 12 subrun(s) of 900. seconds each This table concerns car-type=2 NUM LINK FLOW SAT LINK DIST. DELAY YNEY TIME STOPS MAX QUEUE ALL TYPES OF NUMBER VEH/H FLOW LGTH TRVLLD SPENT VEH.H/H TIME VEH/H VEH SUB AV SD MIN MAX VH/H MTRS VH.KM/H AV SD MIN MAX SECS VEH.H/H AV SD MIN MAX % CAP AV SD MIN MAX >----< ---< >---->------< >-----1 56 .00 .00 15.0 12 2114 60 64 1800 150 9.00 .00 .00 .25 0 0 0 0 0 25 0 0 0 0 12 2113 60 2 56 64 1800 150 9.05 .03 .00 .03 .09 15.0 .28 12 6 25 10 4 4 0 6 2 23 NUM TOTAL TOTAL TOTAL TOTAL SPEED OF DISTANCE TIME DELAY STOPS SUB TRAVELLED SPENT VEH.KM/H VEH.H/H VEH.H/H VEH/H KM/H 12 45.08 1.95 .39 49. 23.16 AVERAGE .79 .11 .10 15. 1.19 STANDARD DEVIATION 44.05 1.81 .24 21.54 MINIMUM 24. 46.75 2.11 .53 72. 24.91 MAXIMUM

# B. Prototype input datasets

### B.1. MANDAT dataset

//MANDAT-dataset for a PROTOTYPE for an isolated RAMP-METERING INSTALLATION //DETector-element-names \_ SPEED\_DETECTOR\_ LONG\_LOOP\_ OCCUPANCY\_DETECTOR\_ SHORT\_LOOP\_ //TIMer-element-names INIT\_TIMER\_ YELLOW\_TIMER\_ GREEN\_TIMER RED\_TIMER\_ METERING\_TIMER\_ MEASURE\_TIMER\_ //MEMcry-element-names PRINT MIN\_GREEN\_TIME MIN\_YELLOW\_TIME\_ MAX\_GREEN\_TIME MAX\_YELLOW\_TIME\_ MIN\_RED\_TIME\_ MIN\_METERING\_TIME\_ MAX\_METERING\_TIME\_ CUR\_METERING\_TIME\_ METERING\_TIME\_ MEASURE\_PERIOD\_ MEASURE\_MET\_TIME\_ PERIOD CLEARANCE\_TIME\_ FLOW\_NEW\_ FLOW\_OLD\_ FLOW\_CUR\_ FLOW\_PER\_HOUR\_ SET\_ON\_FLOW\_ SET\_OFF\_FLOW\_ SPEED NEW SPEED OLD SPEED\_CUR VEH\_SPEED\_OLD VEH SPEED CUR SPEED UPSTREAM SPEED DOWNSTREAM SET\_MAX\_ON\_SPEED\_ SET\_MAX\_OFF\_SPEED\_ SET\_OFF\_SPEED\_ SET\_ON\_SPEED\_ ALPHA\_INCREASE\_FLOW\_ ALPHA\_DECREASE\_FLOW\_ BETA\_INCREASE\_SPEED\_ BETA\_DECREASE\_SPEED\_ ALPHA\_ BETA\_ CAPACITY OCCUPANCY CUR OCCUPANCY\_SETPOINT ALINEA\_CONSTANT\_ ONRAMP\_VOLUME\_ //LOGical-element-names \_YELLOW\_ DP \_RED\_ \_GREEN\_ \_DV\_ \_DA\_ \_DK\_ \_MAX\_GREEN\_ \_MIN\_GREEN\_ \_MIN\_YELLOW\_ \_MAX\_YELLOW\_ \_VEHICLE\_DEPARTED MIN\_RED \_METERING\_ \_PROC\_ METER\_ \_CLEARED\_ DUMMY ALTERNATE SPEED\_LEVEL\_ \_\_\_\_\_R₩S\_\_ ALINEA\_ //COLLECTION-names METERING\_INSTALL\_ METERING\_SIGNAL\_ //\$\$\$/00 statements on intersection level S(PERIOD\_=3600/MEASURE\_PERIOD\_) .= S(INIT\_TIMER\_=0) .= S(INIT\_TIMER\_=0)
.= S(INIT\_TIMER\_=0) S(MEASURE TIMER =0) S(CUR\_METERING\_TIME\_=MIN\_METERING\_TIME\_) S\_METERING\_,S(METERING\_TIMER\_=0) .= S\_GREEN\_\$\$ E\_METERING .= S(METERING\_TIMER\_=CUR\_METERING\_TIME\_)

page -32-

//\$\$\$/\$\$ statements on signal level

S\_CLEARED\_ .= S(INIT\_TIMER\_=0) S\_RED\_ .= S(INIT\_TIMER\_=0) E\_MIN\_GREEN\_ .= S(INIT\_TIMER\_=0) .= S\_GREEN\_ E\_MIN\_GREEN\_ .= S(GREEN\_TIMER\_=MIN\_GREEN\_TIME\_) E\_MAX\_GREEN\_ .= S(GREEN\_TIMER\_=MIN\_GREEN\_TIME\_) S\_MIN\_YELLOW\_, S\_MAX\_YELLOW\_, S(YELLOW\_TIMER\_=0) .= S\_YELLOW\_ E\_MIN\_YELLOW\_ .= S(YELLOW\_TIMER\_=MIN\_YELLOW\_TIME\_) E\_MAX\_YELLOW\_ .= S(YELLOW\_TIMER\_=MIN\_YELLOW\_TIME\_) S\_MIN\_RED\_, S(RED\_TIMER\_=0) .= S\_RED\_ E\_MIN\_RED\_ .= S(RED\_TIMER\_=MIN\_RED\_TIME\_)

//END

### B.2. CONDAT dataset

//CONDAT-dataset for a PROTOTYPE of an isolated RAMP-METERING INSTALLATION, with the RWS and ALINEA control strategy.

//000 /METERING INSTALL : 001 //001 /METERING\_SIGNAL\_00: 01 05 45 /RUS = T /ALINEA = F /CAPACITY = 4400 /ALINEA\_CONSTANT\_ **= 70** /OCCUPANCY\_SETPOINT\_ = 0.18 /ONRAMP\_VOLUME\_ = 800 /MEASURE\_MET\_TIME\_ = 1.0 /SET\_MAX\_ON\_SPEED\_ = 25 /SET\_MAX\_OFF\_SPEED\_ = 45 /MIN\_METERING\_TIME\_ /MAX\_METERING\_TIME\_ = 4.5 = 12.0 /MEASURE\_PERIOD\_ = 30.0/ALPHA\_INCREASE\_FLOW\_ /ALPHA\_DECREASE\_FLOW\_ = 0.25 = 0.25 /BETA\_INCREASE\_SPEED\_ = 0.2 /BETA\_DECREASE\_SPEED\_ = 0.1 /SET\_ON\_FLOW = 3500 /SET\_OFF\_FLOW = 3000 /SET\_ON\_SPEED = 50.0 /SET\_OFF\_SPEED\_ = 70.0 /CLEARANCE\_TIME\_05 = 2.0 /CLEARANCE\_TIME\_45 /MIN\_GREEN\_TIME\_05 = 2.0 = 1.0 /MIN\_GREEN\_TIME\_45 = 1.0 /MAX\_GREEN\_TIME\_05 = 5.0 /MAX\_GREEN\_TIME\_45 = 5.0 /MAA\_GREEN\_TIME\_45 /MIN\_YELLOW\_TIME\_05 /MIN\_YELLOW\_TIME\_45 /MAX\_YELLOW\_TIME\_05 /MAX\_YELLOW\_TIME\_45 = 0.5 = 0.5 = 2.0 = 2.0 /MIN\_RED\_TIME\_05 = 2.0 /MIN\_RED\_TIME\_45 = 2.0 \_DV\_011 = (SPEED\_DETECTOR\_011>0) DV\_012 = (SPEED\_DETECTOR\_012>0) \_DV\_013 = (SPEED\_DETECTOR\_013>0) \_DV\_014 = (SPEED\_DETECTOR\_014>0) \_DV\_015 = (SPEED\_DETECTOR\_015>0) \_DV\_016 = (SPEED\_DETECTOR\_016>0) \_DK\_051 = (SHORT\_LOOP\_051>0) \_DK\_052 = (SHORT\_LOOP\_052>0) \_DK\_053 = (SHORT\_LOOP\_053>0) [DP]054 = (LONG][OOP]054>0)\_DK\_055 = (SHORT\_LOOP\_055>0) \_DK\_451 = (SHORT\_LOOP\_451>0) \_DK\_452 = (SHORT\_LOOP\_452>0)  $DK_{453} = (SHORT_LOOP_{453>0})$ 

S(FLOW\_NEW\_016=FLOW\_NEW\_016+1) .= S\_DV\_011+S\_DV\_012 S(FLOW\_NEW\_014=FLOW\_NEW\_014+1) .= S\_DV\_013+S\_DV\_014

S(FLOW\_NEW\_052=FLOW\_NEW\_052+1) .= S\_DK\_055 + S\_DK\_453 S(FLOW\_NEW\_051=FLOW\_NEW\_051+1) .= S\_DK\_052 + S\_DK\_451 S(VEH\_SPEED\_CUR\_013=SPEED\_DETECTOR\_013) .= S\_DV\_013 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_013.(VEH\_SPEED\_CUR\_013>VEH\_SPEED\_OLD\_013) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_013.(VEH\_SPEED\_CUR\_013<VEH\_SPEED\_OLD\_013) S(VEH SPEED OLD 013=BETA \*VEH SPEED CUR 013+(1-BETA )\*VEH SPEED OLD 013) .= S DV 013 S(VEH\_SPEED\_CUR\_014=SPEED\_DETECTOR\_014) .= S\_DV\_014 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_014.(VEH\_SPEED\_CUR\_014>VEH\_SPEED OLD 014) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_014.(VEH\_SPEED\_CUR\_014<VEH\_SPEED\_OLD\_014) S(VEH\_SPEED\_OLD\_014=BETA\_\*VEH\_SPEED\_CUR\_014+(1-BETA\_)\*VEH\_SPEED\_OLD\_014) .= S\_DV\_014 S(VEH\_SPEED\_CUR\_015=SPEED\_DETECTOR\_015) .= S\_DV\_015 S(VEH\_SPEED\_OLD\_015=BETA\_\*VEH\_SPEED\_CUR\_015+(1-BETA\_)\*VEH\_SPEED\_OLD\_015) .= S\_DV\_015 S(VEH\_SPEED\_CUR\_016=SPEED\_DETECTOR\_016) .= S\_DV\_016 S(BETA\_=BETA\_INCREASE\_SPEED\_) .= S\_DV\_016.(VEH\_SPEED\_CUR\_016>VEH\_SPEED\_OLD\_016) S(BETA\_=BETA\_DECREASE\_SPEED\_) .= S\_DV\_016.(VEH\_SPEED\_CUR\_016<VEH\_SPEED\_OLD\_016) S(VEH\_SPEED\_OLD\_016=BETA\_\*VEH\_SPEED\_CUR\_016+(1-BETA\_)\*VEH\_SPEED\_OLD\_016) .= S\_DV\_016 S\_PROC\_,E\_PROC\_,S(MEASURE\_TIMER\_=0) .= S(MEASURE\_TIMER\_=MEASURE\_PERIOD\_) S(ALPHA\_=ALPHA\_DECREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_016<FLOW\_OLD\_016) S(FLOW\_OLD\_016=ALPHA \*FLOW\_NEW 016+(1-ALPHA\_)\*FLOW OLD 016) .= S PROC S(FLOW\_CUR\_016=FLOW\_NEW\_016),S(FLOW\_NEW\_016=0) .= S\_PROC\_ S(ALPHA\_=ALPHA\_INCREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_014>FLOW\_OLD\_014) S(ALPHA\_=ALPHA\_DECREASE\_FLOW\_) .= S\_PROC\_.(FLOW\_NEW\_014<FLOW\_OLD\_014) S(FLOW\_OLD\_014=ALPHA\_\*FLOW\_NEW\_014+(1-ALPHA\_)\*FLOW\_OLD\_014) .= S\_PROC S(FLOW\_CUR\_014=FLOW\_NEW\_014), S(FLOW\_NEW\_014=0) .= S PROC S(FLOW\_CUR\_052=FLOW\_NEW\_052), S(FLOW\_NEW\_052=0) .= S\_PROC\_ S(FLOW\_CUR\_051=FLOW\_NEW\_051), S(FLOW\_NEW\_051=0) .= S\_PROC\_ S(SPEED\_UPSTREAM\_=(VEH\_SPEED\_OLD\_013+VEH\_SPEED\_OLD\_014)/2) .= S PROC S(SPEED\_DOWNSTREAM\_=(VEH\_SPEED\_OLD\_015+VEH\_SPEED\_OLD\_016)/2) .= S\_PROC\_ S(FLOW\_PER\_HOUR\_ = FLOW\_OLD\_014 \* PERIOD\_) .= S\_PROC\_ S\_METER\_, .= S\_PROC\_.((SPEED\_UPSTREAM\_ < SET\_ON\_SPEED\_)+(SPEED\_DOWNSTREAM\_ < SET\_ON\_SPEED\_)+</pre> +(FLOW\_PER\_HOUR\_> SET\_ON\_FLOW\_)) E\_METER\_ .= S\_PROC\_.(SPEED\_UPSTREAM\_ > SET\_OFF\_SPEED\_).(SPEED\_DOWNSTREAM\_ > SET\_OFF\_SPEED\_). .(FLOW\_PER\_HOUR\_ < SET\_OFF\_FLOW\_).\_DP\_054N S\_SPEED\_LEVEL\_ .= S\_PROC\_.((SPEED\_UPSTREAM\_<SET\_MAX\_ON\_SPEED\_)+(SPEED\_DOWNSTREAM\_<SET\_MAX\_ON\_SPEED\_))
E\_SPEED\_LEVEL\_ .= S\_PROC\_.(SPEED\_UPSTREAM\_>SET\_MAX\_OFF\_SPEED\_).(SPEED\_DOWNSTREAM\_>SET\_MAX\_OFF\_SPEED\_)

S(ONRAMP\_VOLUME\_=CAPACITY\_-FLOW\_PER\_HOUR\_)

page -35-

page -36-

S(OCCUPANCY\_CUR\_=(OCCUPANCY\_DETECTOR\_011 + OCCUPANCY\_DETECTOR\_012)/2) .= S\_PROC\_.ALINEA\_ S(ONRAMP\_VOLUME\_=ONRAMP\_VOLUME\_+ALINEA\_CONSTANT\_\*(OCCUPANCY\_SETPOINT\_-OCCUPANCY\_CUR\_)) .= S\_PROC\_.ALINEA\_ S(METERING TIME =3600/ONRAMP VOLUME ) .= S PROC S(METERING\_TIME\_=MIN(METERING\_TIME\_,MAX\_METERING\_TIME\_)) S(METERING\_TIME\_=MAX(METERING\_TIME\_,MIN\_METERING\_TIME\_)) S(METERING\_TIME\_=MAX\_METERING\_TIME\_) .= S\_PROC\_ .= S\_PROC\_ .= S\_PROC\_.RWS\_.\_SPEED\_LEVEL\_ S\_DUMMY\_012, S(CUR\_METERING\_TIME\_=CUR\_METERING\_TIME\_+MEASURE\_MET\_TIME\_) .= (E\_RED\_05 + E\_RED\_45). .(METERING\_TIME >(CUR\_METERING\_TIME\_+MEASURE\_MET\_TIME\_)).(DK\_053+DK\_452) s\_dummy\_012,s(cur\_metering\_time\_=cur\_metering\_time\_-measure\_met\_time\_) .= (E\_RED\_05 + E\_RED\_45). .(METERING\_TIME\_<(CUR\_METERING\_TIME\_-MEASURE\_MET\_TIME\_)).(\_DK\_053+\_DK\_452) S(CUR\_METERING\_TIME\_=METERING\_TIME\_) .= (E\_RED\_05 + E\_RED\_45).\_DUMMY\_012N E\_DUMPIY\_012 .= E\_RED\_05 + E\_RED\_45 S\_GREEN 45 .= E\_RED\_45 E\_CLEARED\_45 .= S\_GREEN\_45 E\_GREEN\_45 .= (E\_MIN\_GREEN\_45.\_VEHICLE\_DEPARTED\_45+S\_DK\_451.\_MAX\_GREEN\_45+E\_MAX\_GREEN\_45) . \_METER\_ + + S METER S\_YELLOW\_45,E\_ALTERNATE\_05,E\_VEHICLE\_DEPARTED\_45 .= E\_GREEN\_45 .= E\_GREEN\_45 . \_DK\_053 E\_YELLOW\_45 .= E\_MIN\_YELLOW\_45 S RED 45 = E\_YELLOW\_45 .= ((S\_DK\_452+S\_DK\_453+E\_METERING\_+E\_MIN\_RED\_45+S\_RED\_05+S\_CLEARED\_05).(\_DK\_452+\_DK\_453) E\_RED\_45 .\_METERING\_N . \_MIN\_RED\_45N . \_RED\_05 . \_CLEARED\_05 . \_ALTERNATE\_45N) . \_METER\_ + E\_METER\_ S VEHICLE\_DEPARTED\_45 .= S\_DK\_451.\_MIN\_GREEN\_45 S\_CLEARED\_45 .= S(RED\_TIMER\_45=CLEARANCE\_TIME\_45) S\_GREEN\_05 .= E\_RED 05 E\_CLEARED 05 .= S\_GREEN\_05 E\_GREEN\_05 .= (E\_MIN\_GREEN\_05.\_VEHICLE\_DEPARTED\_05+S\_DK\_052.\_MAX\_GREEN\_05+E\_MAX\_GREEN\_05).\_METER\_+ + S METER S\_YELLOW\_05,E\_ALTERNATE\_45,E\_VEHICLE\_DEPARTED\_05 .= E\_GREEN\_05 S\_ALTERNATE\_05 .= E\_GREEN\_05 . .= S\_DK\_051 . \_MTN\_YELLOW\_05N + E\_MAX\_YELLOW\_05 .= E\_YELLOW\_05 DK 452 E\_YELLOW\_05 S\_RED\_05 .≈ ((S\_DK\_053 + E\_METERING\_ + E\_MIN\_RED\_05 + S\_RED\_45 + S\_CLEARED\_45) . \_DK\_053 . \_METERING\_N E\_RED\_05 . \_MIN\_RED\_OSN . \_RED\_45 . \_CLEARED\_45 . \_ALTERNATE\_O5N) . \_METER\_ + E\_METER\_ .= S\_DK\_052.\_MIN\_GREEN\_05 S VEHICLE DEPARTED 05 S\_CLEARED\_05 .= S(RED\_TIMER\_05=CLEARANCE\_TIME\_05)

//END