**Memo:**

Van: Marcel Fick

Aan: Cyril Cappendijk

Onderwerp: Traffick2TLCGen

**Inleiding**

Ik heb uitvoerig de gegenereerde code bekeken en overal waar ik verschillen heb gevonden met mijn eigen versie heb ik tot in detail beschreven hoe ik denk dat het aangepast moet worden. Daarnaast heb ik de “standaard”kruising opnieuw gebouwd en daar ook in de REG en PRIO file de wijzigingen doorgevoerd zodat met file compare en wat knip en plak werk snel de aanpassingen kunnen worden doorgevoerd. Let op ook de file prio.c heb ik aangepast.

De komende weken ga ik verder met de “minder” ingewikkelde items van Traffick, denk bijvoorbeeld aan alles wat DVM te maken heeft. Wordt dus nog wel vervolgt.

**TLCGen invoer**

Het lukt om kruispuntarmen te definiëren, maar vervolgens kan ik geen kruispuntarmen toedelen aan fasecycli. Als ik vervolgens TLCGen afsluit en opnieuw opstart lukt dat laatste wel.

**Genereren REG File**

**define’s na Versie commentaar**

#define TRAFFICK toevoegen als Traffick is aangevinkt in TLCGen.

**init\_application() voor aanroep post\_init\_application()**

1.

Het aantal definitie\_harde\_koppeling() is juist, maar de inhoud van de functie aanroep niet. Voor alle definities wordt een regel gegeneerd voor een koppeling van fc02 op fc62.

(het type koppeling kan worden herkend door een koppeling van einde (verleng)groen)

Is het een koppeling vanaf einde groen dan is het 8e veld in de functie aanroep gelijk aan TRUE.

Is de koppeling vanaf einde verlenggroen dan is het 8e veld in de functie aanroep gelijk aan FALSE.

Het 9e veld in de functie aanroep geeft aan of de volgrichting op basis van eigen detectie mag realiseren onafhankelijk van de voedende richting. Dit kun je in TLCGen niet opgeven. Mooi als dat wel zou kunnen anders de volgende default instelling toepassen.

Volgrichting is fiets of voetganger, dan default FALSE. In alle andere gevallen default TRUE.

Het 10e veld in de functie aanroep is de maximale verlengduur nadat de harde koppeling actief is geweest. Dit zou instelbaar moeten zijn door middel van een apart tijdelement. Die is nu echter (nog)niet configureerbaar in TLCGen.

2.

Het aantal definitie\_vtg\_gescheiden() is niet juist. (2x te hoog) Het lijkt erop dat voor iedere voetgangerskoppeling een regel wordt gegenereerd, maar één definitie bevat de informatie van twee voetgangerskoppelingen, bijv. van fc31 naar fc32 en omgekeerd (dus ook van fc32 naar fc31)

(het type koppeling kan worden “herkend” doordat een koppeling twee kanten op is gedefinieerd, zoals in het voorbeeld van fc31 en fc32. Bovendien is er in dat geval altijd sprake van een naloop vanaf startgroen en nooit vanaf eindegroen)

De inhoud van de functie aanroep is niet juist. Voor alle definities wordt een regel gedefinieerd voor een koppeling van fc31 op fc32 en omgekeerd)

3.

Het aantal definitie\_gelijkstart\_lvk() is juist, maar de inhoud van de functie aanroep niet. Voor alle definities wordt een regel gegeneerd voor een gelijkstart tussen fc22 op fc32.

(het type koppeling kan worden herkend door een gelijkstart van het type geen deelconflict welke in twee richtingen is gedefinieerd, dus fc22 met fc32 en fc32 met fc22)

Let op: als richtingen van dezelfde oversteek in meerdere gelijkstarten voorkomt dan moet er één functie aanroep gemaakt worden. In het voorbeeld kruispunt zou dit het geval zijn indien er ook een gelijkstart van het type geen deelconflict zou zijn tussen fc24 en fc84.

In dat geval zou gegenereerd moeten worden:

definitie\_gelijkstart\_lvk(fc24, fc34, fc33, fc84);

in plaats van:

definitie\_gelijkstart\_lvk(fc24, fc84, NG, NG);

definitie\_gelijkstart\_lvk(fc24, fc34, NG, NG);

definitie\_gelijkstart\_lvk(fc33, fc84, NG, NG);

4.

Het aantal definitie\_voorstart\_dcf() is juist, maar de inhoud van de functie aanroep niet. Voor alle definities wordt een regel gegeneerd voor een voorstart van fc22 op fc05.

Opmerking:

Definitie voor twee richtingen met een gelijkstart type deelconflict moet ik nog uitwerken.

**PreApplication() voor aanroep PreApplication\_Add()**

1.

Toevoegen voor aanroep traffick2tlcgen\_detectie():

/\* bijwerken kruispunt variabelen \*/

/\* ------------------------------ \*/

traffick2tlcgen\_kruispunt();

2.

De functie aanroep traffick2tlcgen\_instel heeft inmiddels één extra parameter. Dit betreft de parameter waarmee het alternatief realiseren per blok instelbaar kan worden gemaakt. Het argument, PRM[parmaltb##], komt in de functie aanroep na SCH[schaltg##]. Indien deze functionaliteit in TLCGen niet is aangevinkt dan volgt op deze plek een NG in de functie aanroep.

**Aanvragen() voor aanroep Aanvragen\_Add()**

Toevoegen voor aanroep Aanvragen\_Add()

/\* Traffick2TLCGen \*/

/\* --------------------- \*/

if (SCH[schtraffick2tlcgen]) fiets\_voorrang\_aanvraag();

**Maxgroen voor aanroep Maxgroen\_Add()**

xyprintf(92,6 … etc. tekst “TFB MTG” wijzigen in “TFB Aled”, één spatie tussen TFB en Aled.

xyprintf(117, 7 … etc. “MTG[fc]” wijzigen in “Aled[fc]”.

**Meetkriterium() voor aanroep Meetkriterium\_Add()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_MVG();

**Synchronisaties() voor aanroep Synchronisaties\_Add()**

Verwijderen:

for (i = 0; i < aantal\_hki\_kop; ++i)

{

count fc1 = hki\_kop[i].fc1; /\* voedende richting \*/

count fc2 = hki\_kop[i].fc2; /\* volg richting \*/

if (PRML[ML][fc1] != PRIMAIR) REAL\_SYN[fc1][fc2] =

REAL\_SYN[fc2][fc1] = FALSE;

}

Toelichting:

Dit is een TLC gen bugfix, die ik nu heb opgenomen in de functie Traffick2TLCgen\_uitstel().

**RealisatieAfhandeling() na aanroep Synchroniseer\_PG()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen])

{

Traffick2TLCgen\_PAR();

BeeindigAltRealisatie();

}

**RealisatieAfhandeling() voor aanroep Alternatief\_Add()**

Toevoegen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen])

{

Traffick2TLCgen\_PAR();

BeeindigAltRealisatie();

}

**RealisatieAfhandeling() na aanroep Alternatief\_Add()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) BugFix\_RR\_bij\_HKI();

**RealisatieAfhandeling() voor aanroep Modules\_Add()**

Toevoegen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) BugFix\_RR\_bij\_HKI();

**PostApplication() voor aanroep PostApplication\_Add()**

Toevoegen:

/\* Traffick2TLCGen \*/

#if (!defined AUTOMAAT && !defined AUTOMAAT\_TEST) || defined

VISSIM || defined PRACTICE\_TEST

if (SCH[schtraffick2tlcgen]) FlightTraffick();

#endif

**System\_application() voor aanroep post\_system\_application()**

Toevoegen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) verklik\_fiets\_voorrang();

**dump\_application() voor aanroep post\_dump\_application()**

Toevoegen:

/\* Traffick2TLCGen \*/

#if (!defined AUTOMAAT && !defined AUTOMAAT\_TEST) || defined

VISSIM || defined PRACTICE\_TEST

if (SCH[schtraffick2tlcgen]) DumpTraffick();

#endif

**Genereren PRIO File**

**define’s na Versie commentaar**

#define TRAFFICK toevoegen als Traffick is aangevinkt in TLCGen.

**AfkapGroenExtra()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO\_TOE();

**StartGroenMomentenExtra()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCpas\_TVG\_aan();

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCzet\_TVG\_terug();

**PrioriteitsOpties(void)**

Als volgt wijzigen:

{

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen])

{

Traffick2TLCgen\_PRIO\_OPTIES();

Traffick2TLCgen\_HLPD\_nal(fcxx, fcyy, T\_max[thdxxyynal);

}

#ifdef PRIO\_ADDFILE

PrioriteitsOpties\_Add();

#endif

#ifdef PRIO\_ADDFILE

PrioriteitsNiveau\_Add();

#endif

}

De aanroep ***Traffick2TLCgen\_HLPD\_nal()*** is lastig. Ik denk dat ik dat generiek kan oplossen in de functie Traffick2TLCgen\_PRIO\_OPTIES. ***De aanroep(en) hier komen dan te vervallen.*** Wat ik wel nodig heb is de mogelijkheid om nalooptijden in te voeren in TLCGen. Van iedere gedefinieerde ARM X naar iedere andere ARM Y moet een nalooptijd gedefinieerd kunnen worden. (NG betekent geen naloop van ARM X naar ARM Y) Naamgeving: tnalhdarmXarmY.

**PrioriteitsToekenningExtra()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) corrigeer\_blokkeringstijd\_OV();

**TegenhoudenConflictenExtra()**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO\_RR();

**PostAfhandelingPrio () – helemaal onderaan in de functie**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO();

**PrioPARCorrecties () – helemaal onderaan in de functie**

Verwijderen:

/\* Traffick2TLCGen \*/

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO\_PAR();

**Aanpassingen PRIO.C**

1.

Toevoegen voor aanroep WachtTijdBewaking\_Add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#ifdef TRAFFICK

if (SCH[schtraffick2tlcgen]) corrigeer\_maximum\_wachttijd\_OV();

#endif

2.

Toevoegen voor aanroep Blokkeringstijd\_Add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#ifdef TRAFFICK

if (SCH[schtraffick2tlcgen]) corrigeer\_blokkeringstijd\_OV();

#endif

3.

Toevoegen voor aanroep PrioriteitsToekenning\_Add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#ifdef TRAFFICK

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO\_TOE();

#endif

4.

Toevoegen voor aanroep AfkapGroen\_Add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#if (defined (TRAFFICK) && defined (NALOPEN))

if (SCH[schtraffick2tlcgen]) Traffick2TLCpas\_TVG\_aan();

#endif

5.

Toevoegen voor aanroep StartGroenMomenten\_add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#if (defined (TRAFFICK) && defined (NALOPEN))

if (SCH[schtraffick2tlcgen]) Traffick2TLCzet\_TVG\_terug();

#endif

6.

Toevoegen voor aanroep PrioTegenhouden\_Add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#ifdef TRAFFICK

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO\_RR();

#endif

7.

Toevoegen voor aanroep PrioAlternatieven\_Add(): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#ifdef TRAFFICK

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO\_PAR();

#endif

8.

Toevoegen voor aanroep PostAfhandelingPrio\_Add (): (juist na #ifdef PRIO\_ADDFILE)

/\* Traffick2TLCGen \*/

#ifdef TRAFFICK

if (SCH[schtraffick2tlcgen]) Traffick2TLCgen\_PRIO();

#endif

**Opnieuw bouwen testkruispunt 123456 (in CCOL 11)**

1.

De schakelaar traffick2tlcgen wordt 2x aangemaakt (zowel in sys als in tab file)

2.

De volgende include files worden niet gegenereerd”

traffick2tlcgen.c

traffick2tlcgen.h

trigvar.c

tx\_synch.h

plevar.c

plevar.h

De laatste vier heb ik niet elders kunnen vinden, dus gegenereerd zonder signaalplanplus.

3.

De functie pre\_msg\_fctiming wordt niet aangemaakt, heb ik (leeg) toegevoegd aan 123456reg.add

4.

Alle wijzigingen uit dit document heb ik doorgevoerd in de volgende files:

k123456reg.c

k123456prio.c

prio.c