Stc - Starter Calculation

1 [Starter Calculation] Starter Calculation

1.1 [Overview]

Figure 1: [Stc Function Overview]

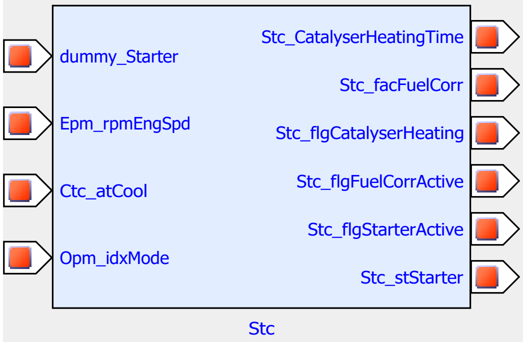
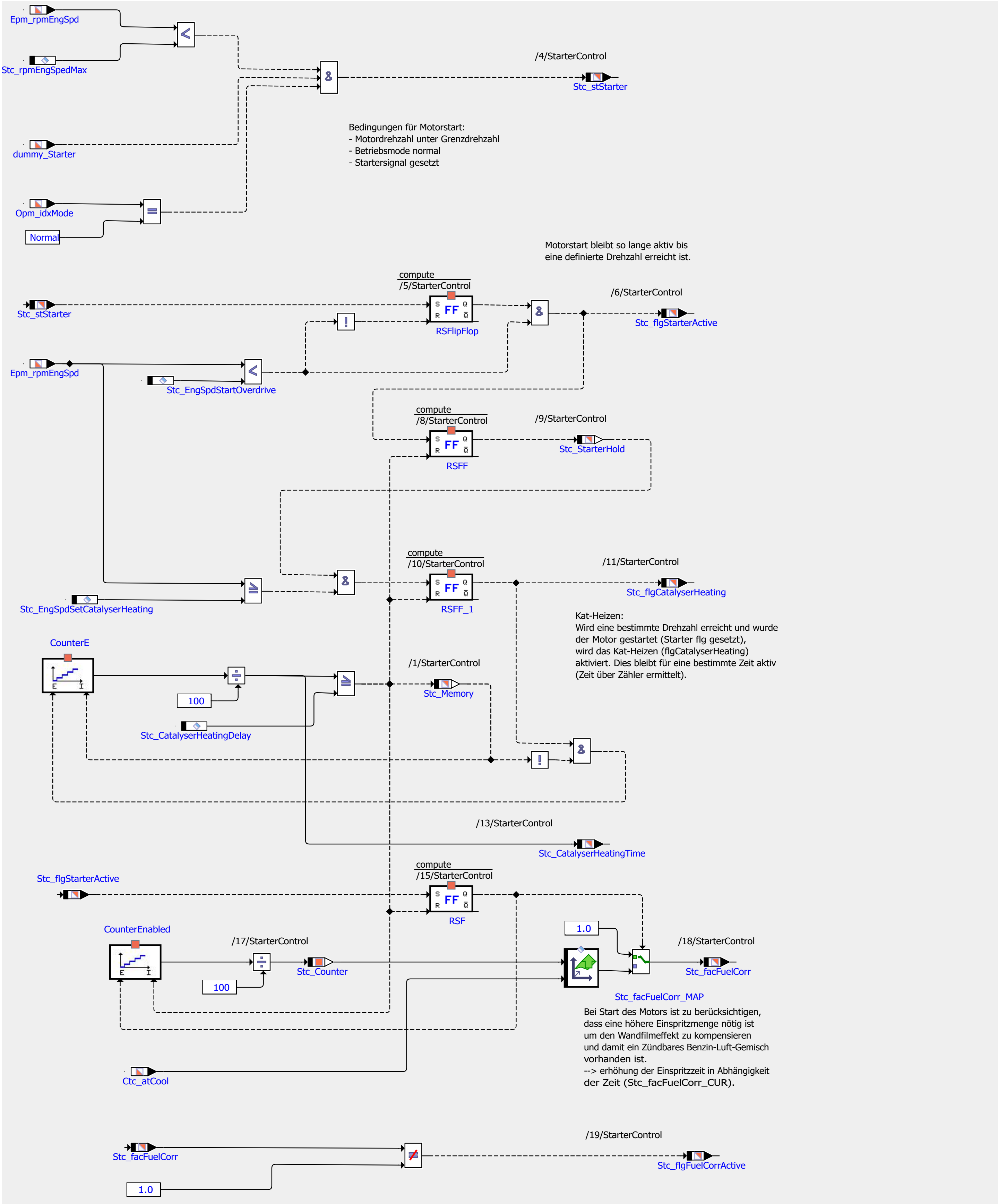


Figure 2: [StarterControl.Main]



Bedingungen für Motorstart:
- Motordrehzahl unter Grenzdrehzahl
- Betriebsmode normal
- Startersignal gesetzt

Motorstart bleibt so lange aktiv bis eine definierte Drehzahl erreicht ist.

Kat-Heizen:
Wird eine bestimmte Drehzahl erreicht und wurde der Motor gestartet (Starter flg gesetzt), wird das Kat-Heizen (flgCatalyserHeating) aktiviert. Dies bleibt für eine bestimmte Zeit aktiv (Zeit über Zähler ermittelt).

Bei Start des Motors ist zu berücksichtigen, dass eine höhere Einspritzmenge nötig ist um den Wandfilmeffekt zu kompensieren und damit ein Zündbares Benzin-Luft-Gemisch vorhanden ist.
--> erhöhung der Einspritzzeit in Abhängigkeit der Zeit (Stc_facFuelCorr_CUR).

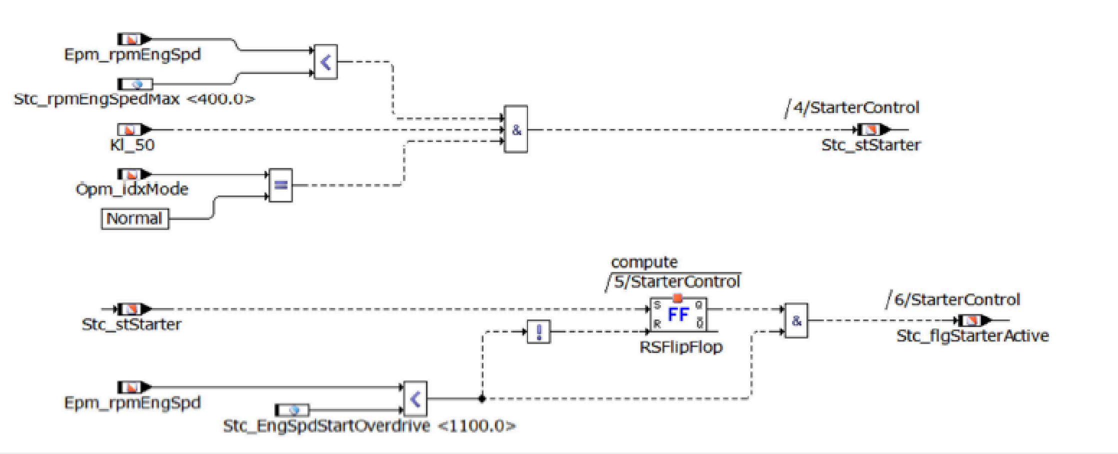
1.2 Starten des Motors [Starten des Motors]

Bei einem Startvorgang eines Verbrennungsmotors wird mit Hilfe der Klemme 50 der Anlasser betätigt. Er schleppt den Motor auf eine Drehzahl bei der er selbstständig auf die Leerlaufdrehzahl beschleunigen kann. Der Startvorgang muss von der Motorsteuerung erkannt werden, damit diese bestimmte Anpassungen vornehmen kann.

Bedingungen für den Motorstart:

- Drehzahl kleiner Stc_rpmEngSpedMax
- Motor in normalem Betriebszustand (Opm_idxMode)
- Startersignal (Klemme 50)

Figure 3: Programmausschnitt Startvorgang [stc_png_1]

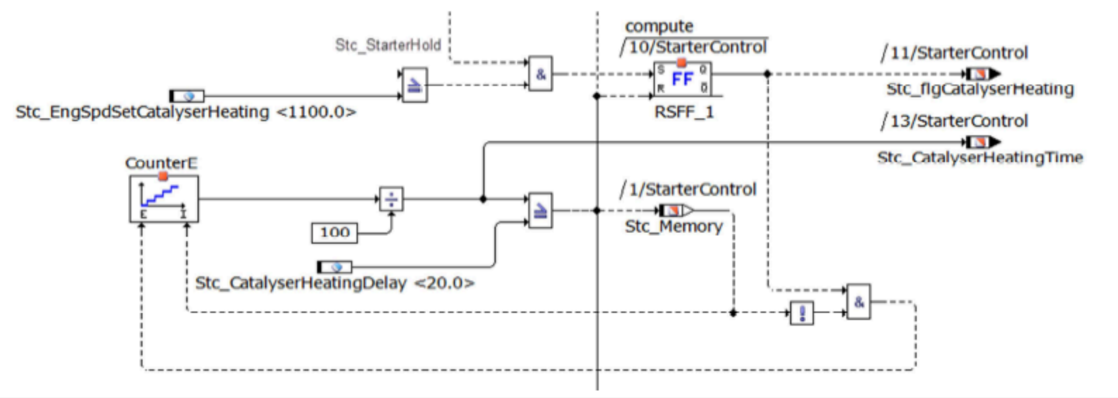


Bei Erkennung eines Startvorgangs wird die variable Stc_stStarter gesetzt. Dadurch wird der Flip Flop Baustein (RSFlipFlop) solange gesetzt, bis eine bestimmte Drehzahl überschritten wird (parametrierbar über Stc_EngSpdStartOverdrive). Durch Setzten dieser Variable wird erstens die Drosselklappenstellung mit dem Parameter T2t_ThrPosStart auf einen während des Startvorgangs festen Wert gestellt. Der Saugrohrdruck sollte dabei nahe dem Umgebungsdruck liegen, um genügend Moment für den Hochlauf des Motors zu erzeugen. Zweitens ist es notwendig, die Einspritzmenge mittels des Parameterkennfeldes Stc_facFuelCorr_MAP im Vergleich zum Normalbetrieb zu erhöhen, um den Wandfilmeffekt auszugleichen. Dieser entsteht dadurch, dass der eingespritzte Kraftstoff nicht komplett verdampft, sondern an den kalten Zylinderwänden kondensiert. Der kondensierte Kraftstoff bleibt unverbrannt, da kein zündfähiges Gemisch vorhanden ist. Je kälter der Motor beim Startvorgang ist, desto höher muss die Einspritzmenge gewählt werden, um den Motor starten zu können. Dies kann die bis zu 8-fache Menge an Kraftstoff erfordern, die bei warmem Motor nötig wäre. Das Kennfeld (Stc_facFuelCorr_MAP) ist abhängig von der Zeit nach Motorstart und der Kühlwassertemperatur des Motors. Eine Simulation des Startvorgangs gegen LABCAR ist in Kapitel 6.1 beschrieben.

1.3 Katalysator heizen nach Motorstart [Katalysator heizen nach Motorstart]

Nach Beendigung des Startvorgangs erfolgt die Übergabe an den Katalysator Heizvorgang. Das ist der Fall, sobald der Parameter Stc_StarterHold gesetzt und die Mindestdrehzahl Stc_EngSpdSetCatalyserHeating erreicht ist. Damit wird der Flip Flop Baustein (RSFF_1) auf 1 gesetzt und bleibt bis zum Ende der Katalysatorheizzeit aktiv. Die Heizzeit wird durch einen Counter berechnet, welcher gesetzt wird, sobald die Katalysatorheizphase aktiv ist. Im Parameter Stc_CatalyserHeatingDelay wird die Länge der Katalysatorheizphase in Sekunden angegeben.

Figure 4: Aktivierung und Steuerung Katalysator heizen [stc_png_2]



Zum Heizen des Katalysators soll die Leerlaufdrehzahl angehoben und der Zündwinkel nach spät gestellt werden. Der Ablauf des Vorgangs wird dabei zeitgesteuert. Zur Erhöhung der Leerlaufdrehzahl wird mit Hilfe des Kennfelds Isc_CatHeatTime2HeatingSpeed Zeitabhängig (Stc_CatalyserHeatingTime) die jeweils parametrierte Soll-Leerlaufdrehzahl ausgegeben. Mittels des Parameters Stc_flgCatalyserHeating wird die übliche Soll-Leerlaufdrehzahl überschrieben.

Der Zündwinkel wird nicht direkt adaptiert, sondern über die Anhebung des "langsamen Saugrohrdrucks". Dadurch wird über die Drosselklappe ein höherer Saugrohrdruck eingeregelt, was zur Folge hat, dass der Zündwinkelwirkungsgrad niedriger und somit der Zündwinkel nach spät gestellt wird. Der Verlauf des Saugrohrdruckes über der Zeit (Stc_CatalyserHeatingTime) wird über das Kennfeld T2t_CatHeatTime2pManiRes gesteuert. Eine Simulation der Funktion des Katalysatorheizen gegen LABCAR ist in Kapitel 6.1 beschrieben.

2 [C-Code Source]

2.1 [Code Listing]

```
/* BEGIN: ASCET REGION "Generation Information" */
/******************************************************************************
 * BEGIN: Generation Information
 *-----------------------------------------------------------------------------
 * Component:................Module
 * Name:....................."StarterControl"
 * Implementation:..........."Impl"
 * Dataset:..................."Data"
 * Specification:............Block Diagram
 * Version:..................<empty String>
 * Library Path:.............."smartm160\Project_SmartM160\Function_Modules"
 *-----------------------------------------------------------------------------
 * Project Name:.............."FlexECU_M160"
 * Project Library Path:....."smartm160\Project_SmartM160\"
 *-----------------------------------------------------------------------------
 * Generation Date:..........03.12.2014
 * Generation Time:..........13:41:34
 *-----------------------------------------------------------------------------
 * ASCET Version:............V6.1.4 RB-DGS 2.3
 * ASCET-MD Version:.........V6.1.4
 * ASCET-RP Version:.........V6.1.4
 * ASCET-SE Version:.........V6.1.4.28.19 CID[610]
 *-----------------------------------------------------------------------------
 * END: Generation Information
 ******************************************************************************/
/* END: ASCET REGION "Generation Information" */

/* BEGIN: ASCET REGION "Project Options" */
/******************************************************************************
 * BEGIN: Project Options "Build"/"Code"
 *-----------------------------------------------------------------------------
 *    Build
 *-----------------------------------------------------------------------------
 * Code Generator:...........Object Based Controller Implementation
 * Compiler:.................Microsoft Visual C++ 2008
 * Operating System:.........GENERIC_OSEK
 * Target:...................ANSI-C
 *-----------------------------------------------------------------------------
 *    Code
 *-----------------------------------------------------------------------------
 * Add Comment with Generation Information for each Component [true]: true
 * Add Comment with Implementation Information for each Assignment Statement [true]: true
 * Add Comment with Specification Source for each Statement [true]:..true
 * Add parenthesis for readability [false]:..........................false
 * Casting [MISRA]:..................................................MISRA
 * Force Parenthesis for Binary Logical Operators [false]:...........false
 * Generate Define Directives for Enum Values [false]:...............false
 * Prefix for Component Names [<empty String>]:......................<empty String>
 * Protected against division by zero [true]:........................true
 * Protected Division against Signed Overflow [true]:................true
 * Protected Vector Indices [true]:..................................true
 *-----------------------------------------------------------------------------
 *    Code.Compiler
 *-----------------------------------------------------------------------------
 * Division truncation direction [Zero (T-division)]:................Zero (T-division)
 * Inline directive [__inline]:......................................__inline
 * Integer Bit Size [32]:............................................32
 * Private directive [static]:.......................................static
 * Public directive [<empty String>]:................................<empty String>
 *-----------------------------------------------------------------------------
 *    Code.FixedPoint
 *-----------------------------------------------------------------------------
 * Allow Double bit Size for Division Numerators [true]:.............true
 * Allow Limit Service for Assignment Limitation [true]:.............true
 * Arithmetic Service Set [<None>]:..................................<None>
 * Generate Limiters (may be changed locally) [true]:................true
 * Generate Round Operation on float to integer Assignment [true]:...true
 * Maximum bit Length (float) [64]:..................................64
 * Maximum bit Length (int) [32]:....................................32
 * Result on Division by Zero [numerator]:...........................numerator
 * Temp Vars always 32 bit (integer) [false]:........................false
 * Use power of 2 approximations of literals [false]:................false
 * Use SHIFT Operation on Signed Values instead of DIV Operation [true]: true
 * Use SHIFT Operation on Signed Values instead of MUL Operation [true]: true
 *-----------------------------------------------------------------------------
 *    Code.Optimizations
 *-----------------------------------------------------------------------------
 * Auto-inline private methods (Smaller code-size - may be changed locally) [false]: false
 * Generate well-formed switch [false]:..............................false
 * Hierarchical Code-Generation (may be changed locally) [false]:....false
 * Initialise history variable with zero [false]:....................false
 * Optimize Direct Access Methods (Multiple Levels) [false]:.........false
 * Optimize Direct Access Methods (One Level) [false]:...............false
 * Optimize Static Actions (Restricted Modelling) [false]:...........false
 * Outline Generated Methods (may be changed locally) [false]:.......false
 *-----------------------------------------------------------------------------
 *    Code.Production
 *-----------------------------------------------------------------------------
 * Add Implementation Definitions [true]:............................true
 * Generate Access Macros for [(variables, messages)]:...............(variables, messages)
 * Generate Access Methods for dT (Alternative: use OS dT directly) [true]: true
 * Generate Data Structures [USELOCAL]:..............................USELOCAL
 * Generate Map File [true]:.........................................true
 * Generate OS Configuration [true]:.................................true
 *-----------------------------------------------------------------------------
 *    Station.Build
 *-----------------------------------------------------------------------------
 * Use Customized Data Type Names [false]:...........................false
 *-----------------------------------------------------------------------------
 * END: Project Options "Build"/"Code"
 ******************************************************************************/
/* END: ASCET REGION "Project Options" */
/* BEGIN: ASCET REGION "ASCET-SE AddOn Options" */
/******************************************************************************
 * BEGIN: ASCET-SE AddOn Options
 * ----------------------------------------------------------------------------
 *     Code
 * ----------------------------------------------------------------------------
 * checkMemSectionVolatility [true]:.................................false
 * checkMultipleSend [false]:........................................false
 * distribVarMemClass ["DISTRRAM"]:..................................."RAM"
 * genAlwaysInitValues [false]:......................................true
 * genLogicElementsAs [PACKED_BITFIELD]:.............................PACKED_BITFIELD
 * genObjList [false]:...............................................false
 * implInfoComments [true]:..........................................true
 * initTaskMemClass ["ASD_INIT_TASK_MEM"]:..........................."ASD_INIT_TASK_MEM"
 * isrMemClass ["ASD_ISR_MEM"]:......................................"ASD_ISR_MEM"
 * mainMemClass ["ASD_EXT_CODE_MEM"]:................................"ASD_EXT_CODE_MEM"
 * optimizeUnusedCode [true]:........................................true
 * paramAsSysConst [false]:..........................................false
 * pragmaMemClassAtDecl [false]:.....................................false
 * pragmaMemClassEnabled [true]:.....................................false
 * referenceMemClass ["REFRAM"]:....................................."RAM"
 * shortNames [false]:...............................................false
 * taskMemClass ["ASD_TASK_MEM"]:...................................."ASD_TASK_MEM"
 * virtualParameterMemClass ["VIRT_PARAM"]:.........................."VIRT_PARAM"
 * ----------------------------------------------------------------------------
 *     Code.Appearance
 * ----------------------------------------------------------------------------
 * braceLineFeed [true]:.............................................true
 * genDate [<undef>]:................................................<undef>
 * genTime [<undef>]:................................................<undef>
 * generateSignatureDecorationComments [true]:.......................true
 * lineFeedPosition [LEFT]:..........................................LEFT
 * maxIdentLength [0]:...............................................40
 * maxRightLength [60]:..............................................60
 * minLeftLength [8]:................................................8
 * preventIndentStructInit [true]:...................................true
 * ----------------------------------------------------------------------------
 *     OS
 * ----------------------------------------------------------------------------
 * Os-Config-C_gen_declaration_alarms [false]:.......................false
 * Os-Config-C_gen_declaration_appmodes [false]:.....................false
 * Os-Config-C_gen_dt_calc [false]:..................................true
 * Os-Config-C_gen_initCOM [false]:..................................false
 * Os-Config-C_gen_inittask [false]:.................................true
 * Os-Config-C_gen_main [false]:.....................................false
 * Os-Config-C_gen_process_container [false]:........................true
 * Os-Config-C_gen_startuphook [false]:..............................false
 * asd_exclusive_area ["ASD_EXCLUSIVE_AREA"]:........................"ASCET_exclusive_area"
 * messageDoInit [false]:............................................false
 * messageExternalMessageCopies [false]:.............................false
 * messageGenOSEKDeclarations [true]:................................false
 * messageIgnoreUsageInInitTask [false]:.............................false
 * messageOverloadInitValues [<undef>]:..............................<undef>
 * messageUsageVariant [OPT_COPY]:...................................NON_OPT_COPY
 * modularMessageUse [false]:........................................false
 * osAppModePattern ["%name%"]:......................................"appmode_%name%"
 * osStartupFunction [<undef>]:......................................<undef>
 * ----------------------------------------------------------------------------
 *     OS.OIL
 * ----------------------------------------------------------------------------
 * OIL-COOP-RESOURCE-name ["ASD_Cooperative_Res"]:...................."ASD_Cooperative_Res"
 * OIL-outputFile ["temp.oil"]:......................................"temp.oil"
 * ----------------------------------------------------------------------------
 *     SERAP
 * ----------------------------------------------------------------------------
 * SERAPRefPageMemoryClass ["SERAP_REF"]:............................"SERAP_REF"
 * SERAPWorkPageMemoryClass ["SERAP_WORK"]:.........................."SERAP_WORK"
 * serap [false]:....................................................false
 * serapEmbedded [true]:.............................................true
 * ----------------------------------------------------------------------------
 *     Virtual Address Tables
 * ----------------------------------------------------------------------------
 * addressTable [true]:..............................................false
 * addressTableMemoryClass ["VATROM"]:..............................."VATROM"
 * ----------------------------------------------------------------------------
 * END: ASCET-SE AddOn Options
 ******************************************************************************/

/* END: ASCET REGION "ASCET-SE AddOn Options" */
/* BEGIN: ASCET REGION "Module Data Definitions" */

/******************************************************************************
 * BEGIN: DEFINITION OF SUBSTRUCT VARIABLE 'Stc_RAM'
 * ----------------------------------------------------------------------------
 * memory class:.................................'RAM'
 * model name:...................................'Stc'
 * data set:.....................................'STARTERCONTROL_IMPL_Data'
 * ---------------------------------------------------------------------------*/
struct STARTERCONTROL_IMPL_RAM_SUBSTRUCT Stc_RAM = {
   /* substruct: Stc_RAM.RSFF (modeled as:'RSFF.Stc') */
   {
      /* struct element:'Stc_RAM.RSFF.status' (modeled as:'status.RSFF.Stc') */
      false
   },
   /* substruct: Stc_RAM.RSFF_1 (modeled as:'RSFF_1.Stc') */
   {
      /* struct element:'Stc_RAM.RSFF_1.status' (modeled as:'status.RSFF_1.Stc') */
      false
   }
};
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF SUBSTRUCT VARIABLE 'Stc_RAM'
 ******************************************************************************/

/******************************************************************************
 * BEGIN: DEFINITION OF COMPONENT VARIABLE 'Stc'
 * ----------------------------------------------------------------------------
 * memory class:.................................'ROM'
 * model name:...................................'Stc'
 * data set:.....................................'STARTERCONTROL_IMPL_Data'
 * ---------------------------------------------------------------------------*/
const struct STARTERCONTROL_IMPL Stc = {
   /* substruct: Stc.RSFF (modeled as:'RSFF.Stc') */
   {
      /* type descriptor pointer 'RSFF_IMPL_RAM' for memory class substruct for 'RAM' */
      &St_RAM.RSFF
   },
   /* substruct: Stc.RSFF_1 (modeled as:'RSFF_1.Stc') */
   {
      /* type descriptor pointer 'RSFF_IMPL_RAM' for memory class substruct for 'RAM' */
      &St_RAM.RSFF_1
   }
};
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF COMPONENT VARIABLE 'Stc'
 ******************************************************************************/

/* END: ASCET REGION "Module Data Definitions" */

/* BEGIN: ASCET REGION "Exported Data Definitions" */
```

```
/*******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Stc_CatalyserHeatingDelay'
 * ----------------------------------------------------------------------------*/
const uint8 Stc_CatalyserHeatingDelay = 20;
   /* min=0.0, max=100.0, ident, limit=yes */
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Stc_CatalyserHeatingDelay'
 *******************************************************************************/


/*******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Stc_EngSpdSetCatalyserHeating'
 * ----------------------------------------------------------------------------*/
const sint16 Stc_EngSpdSetCatalyserHeating = 2200;
   /* min=0.0, max=16383.5, fac_2, limit=yes */
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Stc_EngSpdSetCatalyserHeating'
 *******************************************************************************/


/*******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Stc_EngSpdStartOverdrive'
 * ----------------------------------------------------------------------------*/
const sint16 Stc_EngSpdStartOverdrive = 2200;
   /* min=0.0, max=16383.5, fac_2, limit=yes */
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Stc_EngSpdStartOverdrive'
 *******************************************************************************/


/*******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Stc_Memory'
 * ----------------------------------------------------------------------------*/
uint8 Stc_Memory = false;
   /* min=0, max=1, Identity, limit=yes */
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Stc_Memory'
 *******************************************************************************/


/*******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Stc_StarterHold'
 * ----------------------------------------------------------------------------*/
uint8 Stc_StarterHold = false;
   /* min=0, max=1, Identity, limit=yes */
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Stc_StarterHold'
 *******************************************************************************/


/*******************************************************************************
 * BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Stc_facFuelCorr_CUR'
 * ----------------------------------------------------------------------------*/
const struct CharTable1_uint32_11_uint16_STARTERCONTROL_IMPL_TYPE Stc_facFuelCorr_CUR = {
      11,
      {
         0, 3, 5, 8, 10, 13, 15, 18, 20, 23, 25
      },
      {
         2000, 2000, 2000, 1900, 1800, 1700, 1500, 1300, 1000, 1000, 1000
      }
   };
   /* result: min=0.0, max=10.0, fac_1000, limit=yes */
   /* x axis: min=0.0, max=4294967295.0, ident */
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF CHARACTERISTIC TABLE 'Stc_facFuelCorr_CUR'
 *******************************************************************************/


/*******************************************************************************
 * BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Stc_facFuelCorr_MAP'
 * ----------------------------------------------------------------------------*/
const struct CharTable2_uint32_11_uint16_6_uint16_STARTERCONTROL_IMPL_TYPE Stc_facFuelCorr_MAP = {
      11,
      6,
      {
         0, 3, 5, 8, 10, 13, 15, 18, 20, 23, 25
      },
      {
         31104, 33664, 34944, 37504, 41344, 46464
      },
      {
         9600, 7200, 4800, 2400, 2040, 1200,
         9600, 7200, 4800, 2400, 2040, 1200,
         8400, 6000, 4200, 2400, 1920, 1200,
         8400, 6000, 4200, 2280, 1800, 1200,
         7200, 5400, 3600, 2160, 1680, 1200,
         6000, 5400, 3600, 2040, 1560, 1200,
         4800, 4800, 2400, 1800, 1440, 1200,
         3600, 3600, 2400, 1560, 1320, 1200,
         2400, 2400, 1200, 1200, 1200, 1200,
         1200, 1200, 1200, 1200, 1200, 1200,
         1200, 1200, 1200, 1200, 1200, 1200
      }
   };
   /* result: min=0.0, max=15.0, fac_1000, limit=no */
   /* x axis: min=0.0, max=4294967295.0, ident */
   /* y axis: min=0.0078125, max=511.9921875, fac_128 */
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF CHARACTERISTIC TABLE 'Stc_facFuelCorr_MAP'
 ******************************************************************************/


/******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Stc_rpmEngSpedMax'
 * ---------------------------------------------------------------------------*/
const sint16 Stc_rpmEngSpedMax = 800;
   /* min=0.0, max=16383.5, fac_2, limit=yes */
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Stc_rpmEngSpedMax'
 ******************************************************************************/

/* END: ASCET REGION "Exported Data Definitions" */

/******************************************************************************
 * BEGIN: DEFINITION OF MESSAGES
 * ----------------------------------------------------------------------------
 * Total size is [bytes]:.........................8
 * ---------------------------------------------------------------------------*/
/* messages of memory class:............................................'RAM' */
/* messages of size [bytes]:................................................2 */
/* modelled as 'Stc_CatalyserHeatingTime' */
uint16 Stc_CatalyserHeatingTime;
/* modelled as 'Stc_facFuelCorr' */
uint16 Stc_facFuelCorr;
/* messages of size [bytes]:................................................1 */
/* modelled as 'Stc_flgCatalyserHeating' */
uint8 Stc_flgCatalyserHeating;
/* modelled as 'Stc_flgFuelCorrActive' */
uint8 Stc_flgFuelCorrActive;
/* modelled as 'Stc_flgStarterActive' */
uint8 Stc_flgStarterActive;
/* modelled as 'Stc_stStarter' */
uint8 Stc_stStarter;
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF MESSAGES
 ******************************************************************************/

#define _CounterE CounterE_Stc
#define _CounterE_REF_ (&(CounterE_Stc))
#define _CounterEnabled CounterEnabled_Stc
#define _CounterEnabled_REF_ (&(CounterEnabled_Stc))
#define _RSF RSF_Stc
#define _RSF_REF_ (&(RSF_Stc))
#define _RSFF Stc.RSFF
#define _RSFF_1 Stc.RSFF_1
#define _RSFF_1_REF_ (&(Stc.RSFF_1))
#define _RSFF_REF_ (&(Stc.RSFF))
#define _RSFlipFlop RSFlipFlop_Stc
#define _RSFlipFlop_REF_ (&(RSFlipFlop_Stc))
#define _Stc_CatalyserHeatingDelay Stc_CatalyserHeatingDelay
#define _Stc_EngSpdSetCatalyserHeating Stc_EngSpdSetCatalyserHeating
#define _Stc_EngSpdStartOverdrive Stc_EngSpdStartOverdrive
#define _Stc_facFuelCorr_CUR Stc_facFuelCorr_CUR
#define _Stc_facFuelCorr_MAP Stc_facFuelCorr_MAP
#define _Stc_facFuelCorr_MAP_REF_ (&(Stc_facFuelCorr_MAP))
#define _Stc_Memory Stc_Memory
#define _Stc_rpmEngSpedMax Stc_rpmEngSpedMax
#define _Stc_StarterHold Stc_StarterHold

/* BEGIN: ASCET REGION "Component Functions" */
/******************************************************************************
 * BEGIN: FUNCTIONS OF COMPONENT
 ******************************************************************************/

/* BEGIN: ASCET REGION "Process Definition 'StarterControl'" */
/******************************************************************************
 * BEGIN: DEFINITION OF PROCESS 'STARTERCONTROL_IMPL_StarterControl'
 * ----------------------------------------------------------------------------
 * model name:...................................'StarterControl'
 * memory class:................................'CODE'
 * ---------------------------------------------------------------------------*/
//#if defined(COMPILE_UNUSED_CODE) || defined(COMPILE_UNUSED__STARTERCONTROL_IMPL_StarterControl)
/* messages used by this process */

/* public StarterControl []  */

void STARTERCONTROL_IMPL_StarterControl (void)
{
   /* temp. variables */
   uint32 _t1uint32;
   uint16 _t1uint16;

   /* define local message copies */
   uint16 Ctc_atCool__STARTERCONTROL_IMPL_StarterControl;
   sint16 Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl;
   uint8 Opm_idxMode__STARTERCONTROL_IMPL_StarterControl;
   uint16 Stc_CatalyserHeatingTime__STARTERCONTROL_IMPL_StarterControl;
   uint16 Stc_facFuelCorr__STARTERCONTROL_IMPL_StarterControl;
   uint8 Stc_flgCatalyserHeating__STARTERCONTROL_IMPL_StarterControl;
   uint8 Stc_flgFuelCorrActive__STARTERCONTROL_IMPL_StarterControl;
   uint8 Stc_flgStarterActive__STARTERCONTROL_IMPL_StarterControl;
   uint8 Stc_stStarter__STARTERCONTROL_IMPL_StarterControl;
   uint8 dummy_Starter__STARTERCONTROL_IMPL_StarterControl;
   /* receive messages implicitly */
   {
      DisableAllInterrupts();
      Ctc_atCool__STARTERCONTROL_IMPL_StarterControl = Ctc_atCool;
      Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl = Epm_rpmEngSpd;
      Opm_idxMode__STARTERCONTROL_IMPL_StarterControl = Opm_idxMode;
      Stc_CatalyserHeatingTime__STARTERCONTROL_IMPL_StarterControl = Stc_CatalyserHeatingTime;
      Stc_facFuelCorr__STARTERCONTROL_IMPL_StarterControl = Stc_facFuelCorr;
      Stc_flgCatalyserHeating__STARTERCONTROL_IMPL_StarterControl = Stc_flgCatalyserHeating;
      Stc_flgFuelCorrActive__STARTERCONTROL_IMPL_StarterControl = Stc_flgFuelCorrActive;
      Stc_flgStarterActive__STARTERCONTROL_IMPL_StarterControl = Stc_flgStarterActive;
      Stc_stStarter__STARTERCONTROL_IMPL_StarterControl = Stc_stStarter;
      dummy_Starter__STARTERCONTROL_IMPL_StarterControl = dummy_Starter;
      EnableAllInterrupts();
   }
   /* StarterControl: sequence call #1  */
   _Stc_Memory = COUNTERE_IMPL_out() / 100U >= _Stc_CatalyserHeatingDelay;
   /* StarterControl: sequence call #2  */
   COUNTERE_IMPL_reset(_Stc_Memory);
   /* StarterControl: sequence call #3  */
   COUNTERE_IMPL_compute(RSFF_IMPL_q(_RSFF_1_REF_) && !_Stc_Memory);
   /* StarterControl: sequence call #4  */
   Stc_stStarter__STARTERCONTROL_IMPL_StarterControl
      = Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl < _Stc_rpmEngSpedMax &&
 dummy_Starter__STARTERCONTROL_IMPL_StarterControl && Opm_idxMode__STARTERCONTROL_IMPL_StarterControl == 1;
   /* StarterControl: sequence call #5  */
   RSFLIPFLOP_IMPL_compute(Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl >= _Stc_EngSpdStartOverdrive,
 Stc_stStarter__STARTERCONTROL_IMPL_StarterControl);
   /* StarterControl: sequence call #6  */
   Stc_flgStarterActive__STARTERCONTROL_IMPL_StarterControl
      = RSFLIPFLOP_IMPL_q()
      && Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl < _Stc_EngSpdStartOverdrive;
   /* StarterControl: sequence call #8  */
   RSFF_IMPL_compute(_RSFF_REF_, COUNTERE_IMPL_out() / 100U >= _Stc_CatalyserHeatingDelay, RSFLIPFLOP_IMPL_q()
      && Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl < _Stc_EngSpdStartOverdrive);
   /* StarterControl: sequence call #9  */
   _Stc_StarterHold = RSFF_IMPL_q(_RSFF_REF_);
   /* StarterControl: sequence call #10  */
   RSFF_IMPL_compute(_RSFF_1_REF_, COUNTERE_IMPL_out() / 100U >= _Stc_CatalyserHeatingDelay, _Stc_StarterHold
      && Epm_rpmEngSpd__STARTERCONTROL_IMPL_StarterControl >= _Stc_EngSpdSetCatalyserHeating);
   /* StarterControl: sequence call #11  */
   Stc_flgCatalyserHeating__STARTERCONTROL_IMPL_StarterControl = RSFF_IMPL_q(_RSFF_1_REF_);
   /* StarterControl: sequence call #13  */
   _t1uint32 = COUNTERE_IMPL_out();
   /* assignment to Stc_CatalyserHeatingTime: min=0, max=10000, hex=100phys+0, limit=(maxBitLength: true,
 assign: true), zero incl.=true  */
   Stc_CatalyserHeatingTime__STARTERCONTROL_IMPL_StarterControl = (uint16)(((_t1uint32 <= 10000U) ? _t1uint32 :
 10000U));
   /* StarterControl: sequence call #14  */
   COUNTEREE_IMPL_compute(RSF_IMPL_q());
   /* StarterControl: sequence call #15  */
   RSF_IMPL_compute(COUNTERE_IMPL_out() / 100U >= _Stc_CatalyserHeatingDelay,
 Stc_flgStarterActive__STARTERCONTROL_IMPL_StarterControl);
   /* StarterControl: sequence call #16  */
   COUNTEREE_IMPL_reset(COUNTERE_IMPL_out() / 100U >= _Stc_CatalyserHeatingDelay);
   /* StarterControl: sequence call #17  */
   _t1uint16
      = ((RSF_IMPL_q()) ? CharTable2_getAt_u32u16u16(_Stc_facFuelCorr_MAP_REF_,COUNTEREE_IMPL_out() /
 100U,Ctc_atCool__STARTERCONTROL_IMPL_StarterControl) : 1000U);
   /* assignment to Stc_facFuelCorr: min=0, max=10000, hex=1000phys+0, limit=(maxBitLength: true, assign: true),
 zero incl.=true  */
```

```
   Stc_facFuelCorr__STARTERCONTROL_IMPL_StarterControl = ((_t1uint16 <= 10000U) ? _t1uint16 : 10000U);
   /* StarterControl: sequence call #18  */
   Stc_flgFuelCorrActive__STARTERCONTROL_IMPL_StarterControl =
 Stc_facFuelCorr__STARTERCONTROL_IMPL_StarterControl != 1000U;
   /* send messages implicitly */
   {
      DisableAllInterrupts();
      Stc_CatalyserHeatingTime = Stc_CatalyserHeatingTime__STARTERCONTROL_IMPL_StarterControl;
      Stc_facFuelCorr = Stc_facFuelCorr__STARTERCONTROL_IMPL_StarterControl;
      Stc_flgCatalyserHeating = Stc_flgCatalyserHeating__STARTERCONTROL_IMPL_StarterControl;
      Stc_flgFuelCorrActive = Stc_flgFuelCorrActive__STARTERCONTROL_IMPL_StarterControl;
      Stc_flgStarterActive = Stc_flgStarterActive__STARTERCONTROL_IMPL_StarterControl;
      Stc_stStarter = Stc_stStarter__STARTERCONTROL_IMPL_StarterControl;
      EnableAllInterrupts();
   }
}
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF PROCESS 'STARTERCONTROL_IMPL_StarterControl'
 ******************************************************************************/
#endif
/* END: ASCET REGION "Process Definition 'StarterControl'" */


/* ****************************************************************************
 * END: FUNCTIONS OF COMPONENT
 ******************************************************************************/
/* END: ASCET REGION "Component Functions" */
```