Opm - Operation Mode

1 [Operation Mode] Operation Mode

1.1 [Overview]

Figure 1: [Opm Function Overview]

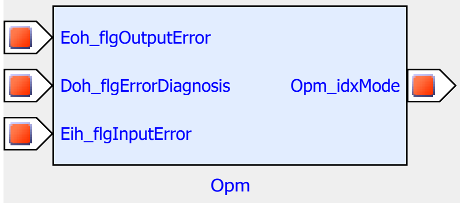
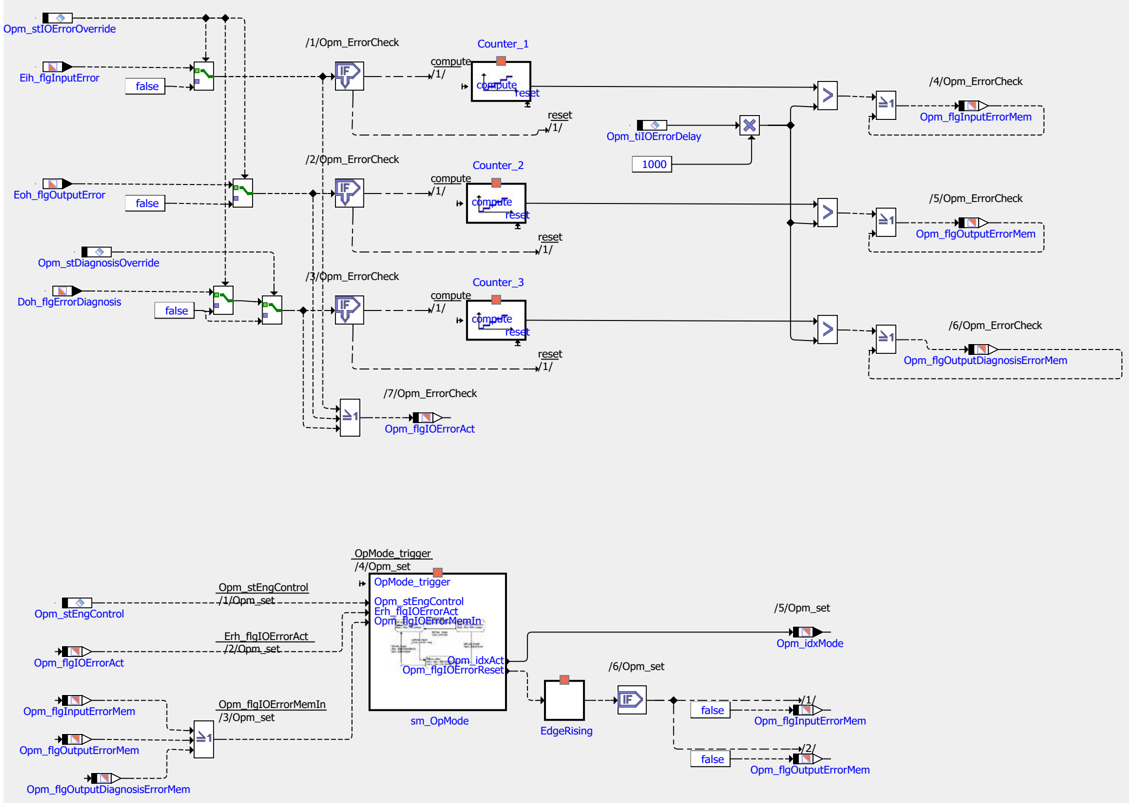


Figure 2: [OperationMode.Main]



1.2 [Operationsmodus]

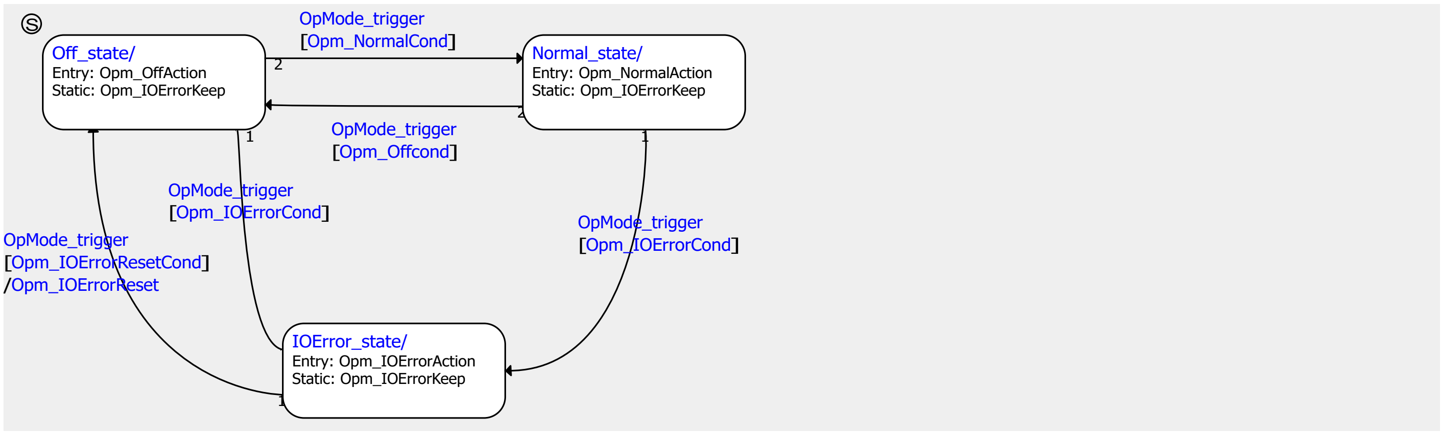
Die Aufgabe des Modul Operationsmodus (Operationmode) besteht in erster Linie darin den Motor im Fehlerfall schnellstmöglich in einen sicheren Zustand zu bringen. Dazu wird:

- die Einspritzung ausgesetzt
- die Zündung ausgeschaltet
- die Drosselklappe in Notlaufposition gebracht

Hierbei wird ein Zustandsautomat (s. Abbildung 5.12) verwendet, der drei Zustände Off, Normal und IOError enthält. Der Zustand Off ist der Startzustand der Motorsteuerung. In diesem Zustand wird die Aktuatorik mit den Werten des sicheren Zustands angesteuert.

Durch setzen des Flags Opm_stEngControl auf true wird in den Zustand Normal auf die Aktuatorik mit den gerechneten Werten der Motorsteuerung oder den manuell überschriebenen Werten versorgt. Beim Auftritt eines Fehlers aus den Sensorik- und Aktuatorik-Modulen wird in den Zustand IOError gewechselt und die Aktuatorik in den sicheren Zustand geschaltet. Erst nach Beseitigung des Fehlers kann mittels Opm_stEngControl = false wieder in den Zustand Off geschaltet werden, um von vorne zu starten. Der Zustandsautomat wird im 100ms-Raster ausgeführt.

Figure 3: sm_OpMode [sm_OpMode.Main]


Off_state/
Entry: Opm_OffAction
Static: Opm_IOErrorKeep
Normal_state/
Entry: Opm_NormalAction
Static: Opm_IOErrorKeep
IOError_state/
Entry: Opm_IOErrorAction
Static: Opm_IOErrorKeep
OpMode_trigger [Opm_NormalCond]
OpMode_trigger [Opm_Offcond]
OpMode_trigger [Opm_IOErrorCond]
OpMode_trigger [Opm_IOErrorCond]
OpMode_trigger [Opm_IOErrorResetCond] /Opm_IOErrorReset

Zur Fehlererkennung werden die eingehenden Signale mit den Fehlerinformationen aus den Komponentenmodulen Eih_InputError und Eoh_OutputError zunächst aufbereitet. Dazu wird im 1ms-Raster geprüft, ob Fehler vorliegen und gegebenenfalls jeweils, getrennt nach Sensorik und Aktuatorik, Zähler inkrementiert. Sobald einer der Zähler den Grenzwert Opm_tiIOErrorDelay (Zeit bis Erkennung in Sekunden) erreicht, wird das entsprechende Flag Opm_flgInputErrorMem oder Opm_flgOutputErrorMem auf true gesetzt und auch nach einem Rücksetzen der Zähler (kein Fehler mehr vorhanden) beibehalten. Der Zustandsautomat verwendet diese Informationen zur Evaluation und wechselt mit Auftreten eines Fehlers am Eingang in den Zustand IOError. In der Variablen Opm_flgIOErrorAct steht der aktuelle Fehlerzustand, also ob aktuell ein Fehler vorhanden ist oder nicht. Sie wird verwendet um zu Prüfen, ob noch ein Fehler vorliegt oder bei Wunsch in den Zustand Off gewechselt werden darf. Mit diesem Wechsel werden dann die Fehlerspeicher Opm_flgInputErrorMem und Opm_flgOutputErrorMem zurückgesetzt.

Die Fehlererkennung kann durch setzen des Flag Opm_stIOErrorOverride auf false komplett ausgeschaltet werden. Das ist jedoch nicht zu empfehlen, da sehr schnell ein unerwartetes Verhalten auftreten kann, das zu Schäden am Motor führen kann, wie beispielsweise einer Sensorstörung bei der Drosselklappe, die zu einem zu großem oder zu kleinem Öffnungswinkel führt1.

Die Vorteile dieser Implementierung sind:

- Durch diese Implementierung kann in einer adäquaten Zeit ein Fehlverhalten erkannt und mit Abschaltung reagiert werden, ohne dabei sofort bei jedem kurzen Störimpuls übertrieben zu reagieren.
- Die Aufteilung nach Sensorik und Aktuatorik ermöglicht es Fehler schneller zu lokalisieren.
- Mit Anpassung der Reaktionszeit bis zur Abschaltung Opm_tiIOErrorDelay kann die „Schärfe“ der Erkennung eingestellt werden.

2 [C-Code Source]

2.1 [Code Listing]

```
/* BEGIN: ASCET REGION "Generation Information" */
/*******************************************************************************
 * BEGIN: Generation Information
 *-----------------------------------------------------------------------------
 * Component:................Module
 * Name:.....................“OperationMode"
 * 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 'Opm_RAM'
 * ---------------------------------------------------------------------------
 * memory class:...............................'RAM'
 * model name:.................................'Opm'
 * data set:...................................'OPERATIONMODE_IMPL_Data'
 * --------------------------------------------------------------------------*/
struct OPERATIONMODE_IMPL_RAM_SUBSTRUCT Opm_RAM = {
   /* substruct: Opm_RAM.Counter_1 (modeled as:'Counter_1.Opm') */
   {
      /* struct element:'Opm_RAM.Counter_1.counter' (modeled as:'counter.Counter_1.Opm') */
      0
   },
   /* substruct: Opm_RAM.Counter_2 (modeled as:'Counter_2.Opm') */
   {
      /* struct element:'Opm_RAM.Counter_2.counter' (modeled as:'counter.Counter_2.Opm') */
      0
   },
   /* substruct: Opm_RAM.Counter_3 (modeled as:'Counter_3.Opm') */
   {
      /* struct element:'Opm_RAM.Counter_3.counter' (modeled as:'counter.Counter_3.Opm') */
      0
   },
   /* substruct: Opm_RAM.EdgeRising (modeled as:'EdgeRising.Opm') */
   {
      /* struct element:'Opm_RAM.EdgeRising.buffer' (modeled as:'buffer.EdgeRising.Opm') */
      false,
      /* struct element:'Opm_RAM.EdgeRising.oldSignal' (modeled as:'oldSignal.EdgeRising.Opm') */
      true
   }
};
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF SUBSTRUCT VARIABLE 'Opm_RAM'
 *****************************************************************************/

/*****************************************************************************
 * BEGIN: DEFINITION OF COMPONENT VARIABLE 'Opm'
 * ----------------------------------------------------------------------------
 * memory class:.................................'ROM'
 * model name:...................................'Opm'
 * data set:.....................................'OPERATIONMODE_IMPL_Data'
 * --------------------------------------------------------------------------*/
const struct OPERATIONMODE_IMPL Opm = {
   /* substruct: Opm.Counter_1 (modeled as:'Counter_1.Opm') */
   {
      /* type descriptor pointer 'COUNTER_IMPL_RAM' for memory class substruct for 'RAM' */
      &Opm_RAM.Counter_1
   },
   /* substruct: Opm.Counter_2 (modeled as:'Counter_2.Opm') */
   {
      /* type descriptor pointer 'COUNTER_IMPL_RAM' for memory class substruct for 'RAM' */
      &Opm_RAM.Counter_2
   },
   /* substruct: Opm.Counter_3 (modeled as:'Counter_3.Opm') */
   {
      /* type descriptor pointer 'COUNTER_IMPL_RAM' for memory class substruct for 'RAM' */
      &Opm_RAM.Counter_3
   },
   /* substruct: Opm.EdgeRising (modeled as:'EdgeRising.Opm') */
   {
      /* type descriptor pointer 'EDGERISING_IMPL_RAM' for memory class substruct for 'RAM' */
      &Opm_RAM.EdgeRising
   }
};
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF COMPONENT VARIABLE 'Opm'
 *****************************************************************************/

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

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

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


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


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


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


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


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


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


/******************************************************************************
 * BEGIN: DEFINITION OF VARIABLE 'Opm_tiIOErrorDelay'
 * ----------------------------------------------------------------------------*/
const uint16 Opm_tiIOErrorDelay = 500;
   /* min=0.0, max=65.535, fac_1000, limit=yes */
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF VARIABLE 'Opm_tiIOErrorDelay'
 ******************************************************************************/

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

/******************************************************************************
 * BEGIN: DEFINITION OF MESSAGES
 * ----------------------------------------------------------------------------
 * Total size is [bytes]:........................1
 * ----------------------------------------------------------------------------*/
/* messages of memory class:..........................................'RAM' */
/* messages of size [bytes]:.............................................1 */
/* modelled as 'Opm_idxMode' */
uint8 Opm_idxMode;
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF MESSAGES
 ******************************************************************************/

#define _Counter_1 Opm.Counter_1
#define _Counter_1_REF_ (&(Opm.Counter_1))
#define _Counter_2 Opm.Counter_2
#define _Counter_2_REF_ (&(Opm.Counter_2))
#define _Counter_3 Opm.Counter_3
#define _Counter_3_REF_ (&(Opm.Counter_3))
#define _EdgeRising Opm.EdgeRising
#define _EdgeRising_REF_ (&(Opm.EdgeRising))
#define _Opm_flgInputErrorMem Opm_flgInputErrorMem
#define _Opm_flgIOErrorAct Opm_flgIOErrorAct
#define _Opm_flgOutputDiagnosisErrorMem Opm_flgOutputDiagnosisErrorMem
#define _Opm_flgOutputErrorMem Opm_flgOutputErrorMem
#define _Opm_stDiagnosisOverride Opm_stDiagnosisOverride
#define _Opm_stEngControl Opm_stEngControl
#define _Opm_stIOErrorOverride Opm_stIOErrorOverride
#define _Opm_tiIOErrorDelay Opm_tiIOErrorDelay
#define _sm_OpMode sm_OpMode_Opm
#define _sm_OpMode_REF_ (&(sm_OpMode_Opm))
```

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

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

/* public Opm_ErrorCheck []  */

void OPERATIONMODE_IMPL_Opm_ErrorCheck (void)
{
   /* define local message copies */
   uint8 Doh_flgErrorDiagnosis__OPERATIONMODE_IMPL_Opm_ErrorCheck;
   uint8 Eih_flgInputError__OPERATIONMODE_IMPL_Opm_ErrorCheck;
   uint8 Eoh_flgOutputError__OPERATIONMODE_IMPL_Opm_ErrorCheck;
   /* receive messages implicitly */
   {
      DisableAllInterrupts();
      Doh_flgErrorDiagnosis__OPERATIONMODE_IMPL_Opm_ErrorCheck = Doh_flgErrorDiagnosis;
      Eih_flgInputError__OPERATIONMODE_IMPL_Opm_ErrorCheck = Eih_flgInputError;
      Eoh_flgOutputError__OPERATIONMODE_IMPL_Opm_ErrorCheck = Eoh_flgOutputError;
      EnableAllInterrupts();
   }
   /* Opm_ErrorCheck: sequence call #1  */
   if (((_Opm_stIOErrorOverride) ? false : Eih_flgInputError__OPERATIONMODE_IMPL_Opm_ErrorCheck))
   {
      /* If-block: sequence call #1/Then #1  */
      COUNTER_IMPL_compute(_Counter_1_REF_);
   }
   else
   {
      /* If-block: sequence call #1/Else #1  */
      COUNTER_IMPL_reset(_Counter_1_REF_);
   } /* end if */
   /* Opm_ErrorCheck: sequence call #2  */
   if (((_Opm_stIOErrorOverride) ? false : Eoh_flgOutputError__OPERATIONMODE_IMPL_Opm_ErrorCheck))
   {
      /* If-block: sequence call #2/Then #1  */
      COUNTER_IMPL_compute(_Counter_2_REF_);
   }
   else
   {
      /* If-block: sequence call #2/Else #1  */
      COUNTER_IMPL_reset(_Counter_2_REF_);
   } /* end if */
   /* Opm_ErrorCheck: sequence call #3  */
   if (((_Opm_stDiagnosisOverride) ? false : ((_Opm_stIOErrorOverride) ? false :
 Doh_flgErrorDiagnosis__OPERATIONMODE_IMPL_Opm_ErrorCheck)))
   {
      /* If-block: sequence call #3/Then #1  */
      COUNTER_IMPL_compute(_Counter_3_REF_);
   }
   else
   {
      /* If-block: sequence call #3/Else #1  */
      COUNTER_IMPL_reset(_Counter_3_REF_);
   } /* end if */
   /* Opm_ErrorCheck: sequence call #4  */
   _Opm_flgInputErrorMem
      = COUNTER_IMPL_out(_Counter_1_REF_) > _Opm_tiIOErrorDelay || _Opm_flgInputErrorMem;
   /* Opm_ErrorCheck: sequence call #5  */
   _Opm_flgOutputErrorMem
      = COUNTER_IMPL_out(_Counter_2_REF_) > _Opm_tiIOErrorDelay || _Opm_flgOutputErrorMem;
   /* Opm_ErrorCheck: sequence call #6  */
   _Opm_flgOutputDiagnosisErrorMem
      = COUNTER_IMPL_out(_Counter_3_REF_) > _Opm_tiIOErrorDelay || _Opm_flgOutputDiagnosisErrorMem;
   /* Opm_ErrorCheck: sequence call #7  */
   _Opm_flgIOErrorAct
      = (((_Opm_stIOErrorOverride) ? false : Eih_flgInputError__OPERATIONMODE_IMPL_Opm_ErrorCheck))
      || (((_Opm_stIOErrorOverride) ? false : Eoh_flgOutputError__OPERATIONMODE_IMPL_Opm_ErrorCheck))
      || (((_Opm_stDiagnosisOverride) ? false : ((_Opm_stIOErrorOverride) ? false :
 Doh_flgErrorDiagnosis__OPERATIONMODE_IMPL_Opm_ErrorCheck)));
}
/* -----------------------------------------------------------------------------
 * END: DEFINITION OF PROCESS 'OPERATIONMODE_IMPL_Opm_ErrorCheck'
 *******************************************************************************/
```

```
#endif
/* END: ASCET REGION "Process Definition 'Opm_ErrorCheck'" */


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

/* public Opm_set []  */

void OPERATIONMODE_IMPL_Opm_set (void)
{
   /* define local message copies */
   uint8 Opm_idxMode__OPERATIONMODE_IMPL_Opm_set;
   /* receive messages implicitly */
   {
      Opm_idxMode__OPERATIONMODE_IMPL_Opm_set = Opm_idxMode;
   }
   /* Opm_set: sequence call #1  */
   SM_OPMODE_IMPL_setOpm_stEngControl(_Opm_stEngControl);
   /* Opm_set: sequence call #2  */
   SM_OPMODE_IMPL_setErh_flgIOErrorAct(_Opm_flgIOErrorAct);
   /* Opm_set: sequence call #3  */
   SM_OPMODE_IMPL_setOpm_flgIOErrorMemIn(_Opm_flgInputErrorMem || _Opm_flgOutputErrorMem ||
 _Opm_flgOutputDiagnosisErrorMem);
   /* Opm_set: sequence call #4  */
   SM_OPMODE_IMPL_OpMode_trigger();
   /* Opm_set: sequence call #5  */
   Opm_idxMode__OPERATIONMODE_IMPL_Opm_set = SM_OPMODE_IMPL_getOpm_idxAct();
   /* Opm_set: sequence call #6  */
   if (EDGERISING_IMPL_compute(_EdgeRising_REF_, SM_OPMODE_IMPL_getOpm_flgIOErrorReset()))
   {
      /* If-block: sequence call #6/Then #1  */
      _Opm_flgInputErrorMem = false;
      /* If-block: sequence call #6/Then #2  */
      _Opm_flgOutputErrorMem = false;
   } /* end if */
   /* send messages implicitly */
   {
      Opm_idxMode = Opm_idxMode__OPERATIONMODE_IMPL_Opm_set;
   }
}
/* ----------------------------------------------------------------------------
 * END: DEFINITION OF PROCESS 'OPERATIONMODE_IMPL_Opm_set'
 ******************************************************************************/
#endif
/* END: ASCET REGION "Process Definition 'Opm_set'" */


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