

Afr - Air Fuel Ratio

1 [Air Fuel Ratio] Air Fuel Ratio

1.1 [Overview]

Figure 1: [Afr Function Overview]

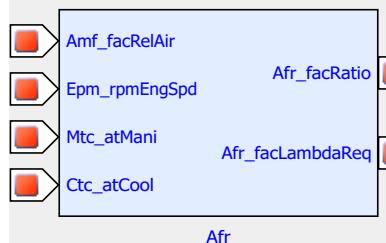
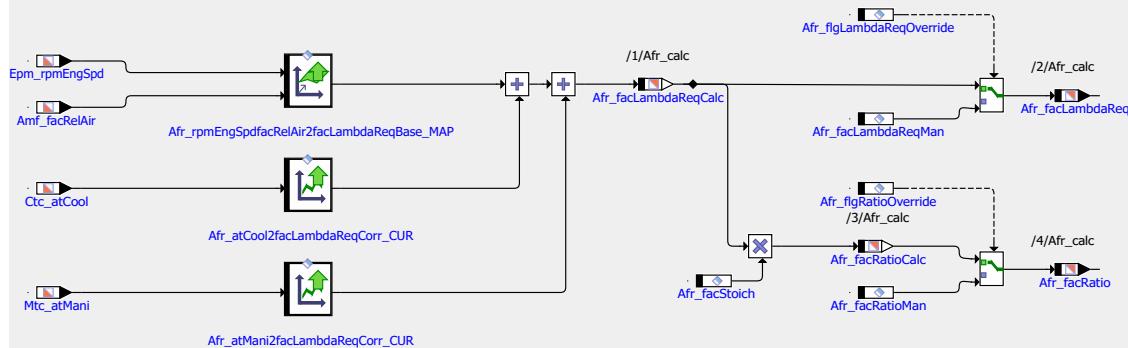


Figure 2: [AirFuelRatio.Main]



1.2 [Luft-Kraftstoff-Verhältnis]

Im Modul Luft-Kraftstoff-Verhältnis (AirFuelRatio) wird auf Basis eines vorgegebenen Lambdawertes das benötigte Verhältnis von Luft zu Kraftstoff berechnet. Dazu wird über das Kennfeld [Afr_rpmEngSpdfacRelAir2facLambdaReqBase_MAP](#) aus der Drehzahl [Epm_rpmEngSpd](#) und der relativen Füllung [Amf_facRelAir](#) ein Basiswert für Lambda ermittelt und anschließend mit Korrekturen beaufschlagt. Die Grundparametrierung aus Erfahrung heraus liegt hier über alle Drehzahlbereiche im Teillastbereich, d. h. bei relativen Füllungen unter 80%, bei $\lambda = 1.0$ und im Volllastbereich über 120% bei $\lambda = 0.7$. Im Übergangsbereich dazwischen findet eine lineare Interpolation statt.

Korrigiert wird zum einen abhängig von der Kühlmitteltemperatur [Ctc_atCool](#), die über die Kennlinie [Afr_atCool2facLambdaReqCorr_CUR](#) eingerechnet wird. Bei kaltem Motor findet eine Gemischanfettung zur Erwärmung des Motors statt, die ab rund 50°C zu Null wird. Zum anderen kann eine Frischlufttemperaturabhängige ([Mtc_atMani](#) über [Afr_atMani2facLambdaReqCorr_CUR](#)) Anfettung oder Abmagerung durchgeführt werden. Dieses Kennfeld ist von Grund auf mit Null bedatet. Der so errechnete Lambdawert kann durch setzen des Flags [Afr_flgLambdaReqOverride](#) manuell mit [Afr_facLambdaReqMan](#) überschrieben werden, bevor er als Nachricht exportiert wird. Gleichung 5.7 zeigt die Berechnungen zum Lambdawert.

Weiterhin wird aus dem Lambdawert durch Multiplikation mit dem stöchiometrischen Faktor [Afr_facStoich](#) das Luft-Kraftstoff-Verhältnis [Afr_facRatio](#) errechnet Gleichung 5.8. Auch hier besteht die Möglichkeit, den errechneten Wert manuell mit [Afr_facRatioMan](#) zu überschreiben (setzen des Flags [Afr_flgRatioOverride](#)), bevor er als Nachricht exportiert wird. Die Auswertung der Funktion [Afr_calc](#) erfolgt im Synchron-Raster.

Figure 3: [afr_jpg_1]

$$\begin{aligned} \lambda_{Soll} &= \lambda_{Basis} \\ &+ \lambda_{Korr, T_{Kühlmittel}} \\ &+ \lambda_{Korr, T_{Frischluft}} \end{aligned}$$

λ_{Soll}	Lambda-Sollwert
λ_{Basis}	Basiswert aus Drehzahl und relativer Füllung
$\lambda_{Korr, T_{Kühlmittel}}$	Korrektur bei kaltem Motor
$\lambda_{Korr, T_{Frischluft}}$	Korrektur für Frischlufttemperatur

Figure 4: [afr_jpg_2]

$$f_{Luft-Kraftstoff} = \lambda_{Soll} \times f_{Stoich}$$

$f_{Luft-Kraftstoff}$	Luft-Kraftstoff-Verhältnis
λ_{Soll}	Lambda-Sollwert
f_{Stoich}	stoichiometrischer Faktor

2 [C-Code Source]

2.1 [Code Listing]

```

/* BEGIN: ASCET REGION "Generation Information" */
/*********************************************************************
* BEGIN: Generation Information
* -----
* Component:.....Module
* Name:....."AirFuelRatio"
* Implementation:....."Impl"
* Dataset:....."Data"
* Specification:.....Block Diagram
* Version:.....<empty String>
* Library Path:....."smartml60\Project_SmartM160\Function_Modules"
* -----
* Project Name:....."FlexECU_M160"
* Project Library Path:...."smartml60\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 ["DISTRAM"]:....."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" */

/*****
* DEFINITION OF COMPONENT VARIABLE OMITTED
* -----
* memory class:.....'ROM'
* model name:.....'Afr'
* reason:.....no local elements
* -----
*/
/* END: ASCET REGION "Module Data Definitions" */

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

/*****
* BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Afr_atCool2facLambdaReqCorr_CUR'
* -----
const struct CharTable1_uint16_10_sint8_AIRFUELRATIO_IMPL_TYPE Afr_atCool2facLambdaReqCorr_CUR = {
    10,
    {
        29824, 34944, 36224, 37504, 38784, 41344, 43904, 46464, 49024, 51584
    },
    {
        -103, -52, -41, -33, -21, -7, 0, 0, 0, 0
    }
};
/* result: min=-0.5, max=0.49609375, fac_256, limit=no */
/* x axis: min=0.0, max=511.9921875, fac_128 */
/* -----
* END: DEFINITION OF CHARACTERISTIC TABLE 'Afr_atCool2facLambdaReqCorr_CUR'
***** */

/*****
* BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Afr_atMani2facLambdaReqCorr_CUR'
* -----
const struct CharTable1_uint16_10_sint8_AIRFUELRATIO_IMPL_TYPE Afr_atMani2facLambdaReqCorr_CUR = {
    10,
    {
        29824, 34944, 36224, 37504, 38784, 41344, 43904, 46464, 49024, 51584
    },
    {
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    }
};
/* result: min=-0.5, max=0.49609375, fac_256, limit=no */
/* x axis: min=0.0, max=511.9921875, fac_128 */
/* -----
* END: DEFINITION OF CHARACTERISTIC TABLE 'Afr_atMani2facLambdaReqCorr_CUR'
***** */

/*****
* BEGIN: DEFINITION OF VARIABLE 'Afr_facLambdaReqCalc'
* -----
uint16 Afr_facLambdaReqCalc = 0;
/* min=0.0, max=2.0, fac_2048, limit=yes */
/* -----
* END: DEFINITION OF VARIABLE 'Afr_facLambdaReqCalc'
***** */

```

```

***** BEGIN: DEFINITION OF VARIABLE 'Afr_facLambdaReqMan'
* -----
const uint16 Afr_facLambdaReqMan = 2048;
/* min=0.7998046875, max=2.0, fac_2048, limit=yes */
* -----
***** END: DEFINITION OF VARIABLE 'Afr_facLambdaReqMan'
***** */

***** BEGIN: DEFINITION OF VARIABLE 'Afr_facRatioCalc'
* -----
uint16 Afr_facRatioCalc = 2048;
/* min=1.0, max=31.99951171875, fac_2048, limit=yes */
* -----
***** END: DEFINITION OF VARIABLE 'Afr_facRatioCalc'
***** */

***** BEGIN: DEFINITION OF VARIABLE 'Afr_facRatioMan'
* -----
const uint16 Afr_facRatioMan = 2048;
/* min=1.0, max=31.99951171875, fac_2048, limit=yes */
* -----
***** END: DEFINITION OF VARIABLE 'Afr_facRatioMan'
***** */

***** BEGIN: DEFINITION OF VARIABLE 'Afr_facStoich'
* -----
const uint8 Afr_facStoich = 235;
/* min=0.0, max=15.9375, fac_16, limit=yes */
* -----
***** END: DEFINITION OF VARIABLE 'Afr_facStoich'
***** */

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

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

***** BEGIN: DEFINITION OF CHARACTERISTIC TABLE 'Afr_rpmEngSpdfacRelAir2facLambdaReqBase_MAP'
* -----
const struct CharTable2_sint16_19_uint16_29_uint8_AIRFUELRATIO_IMPL_TYPE Afr_rpmEngSpdfacRelAir2facLambdaReqBase
_MAP =
{
    19,
    29,
    {
        1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 7000, 8000,
        9000, 10000, 11000, 12000, 13000, 14000
    },
    {
        0, 26, 51, 77, 102, 128, 154, 205, 256, 307, 358, 410, 461, 512, 563, 614, 666,
        717, 768, 819, 870, 922, 973, 1024, 1075, 1126, 1178, 1229, 1280
    },
    {
        128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
        128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
        128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
        128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128,
        128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128, 128
}
;

```



```

/* public Afr_calc [] */

void AIRFUEL RATIO _IMPL_Afr_calc (void)
{
    /* temp. variables */
    uint16 _tluint16;
    sint16 _tlsint16;
    sint16 _t2sint16;
    uint32 _tluint32;

    /* define local message copies */
    uint16 Afr_facLambdaReq__AIRFUEL RATIO _IMPL_Afr_calc;
    uint16 Afr_facRatio__AIRFUEL RATIO _IMPL_Afr_calc;
    uint16 Amf_facRelAir__AIRFUEL RATIO _IMPL_Afr_calc;
    uint16 Ctc_atCool__AIRFUEL RATIO _IMPL_Afr_calc;
    sint16 Epm_rpmEngSpd__AIRFUEL RATIO _IMPL_Afr_calc;
    uint16 Mtc_atMani__AIRFUEL RATIO _IMPL_Afr_calc;
    /* receive messages implicitly */
    {

        DisableAllInterrupts();
        Afr_facLambdaReq__AIRFUEL RATIO _IMPL_Afr_calc = Afr_facLambdaReq;
        Afr_facRatio__AIRFUEL RATIO _IMPL_Afr_calc = Afr_facRatio;
        Amf_facRelAir__AIRFUEL RATIO _IMPL_Afr_calc = Amf_facRelAir;
        Ctc_atCool__AIRFUEL RATIO _IMPL_Afr_calc = Ctc_atCool;
        Epm_rpmEngSpd__AIRFUEL RATIO _IMPL_Afr_calc = Epm_rpmEngSpd;
        Mtc_atMani__AIRFUEL RATIO _IMPL_Afr_calc = Mtc_atMani;
        EnableAllInterrupts();
    }

    /* Afr_calc: sequence call #1 */
    _tluint16
    =
    (uint16)
    ((uint16)CharTable2_getAt_s16u16u8(_Afr_rpmEngSpdfacRelAir2facLambdaReqBase_MAP_REF_,Epm_rpmEngSpd__AIRFUEL RATIO _IMPL_
    << 1);
    _tlsint16
    =
    (sint16)_tluint16 +
    CharTable1_getAt_u16s8(_Afr_atCool2facLambdaReqCorr_CUR_REF_,Ctc_atCool__AIRFUEL RATIO _IMPL_Afr_calc) +
    CharTable1_getAt_u16s8(_Afr_atMani2facLambdaReqCorr_CUR_REF_,Mtc_atMani__AIRFUEL RATIO _IMPL_Afr_calc);
    /* assignment to Afr_facLambdaReqCalc: min=0, max=4096, hex=2048phys+0, limit=(maxBitLength: true, assign: true), zero incl.=true */
    _Afr_facLambdaReqCalc
    =
    (_tlsint16 >= 0) ? (((uint16)_tlsint16 <= 512U) ? (_t2sint16 = _tlsint16 << 3 , (uint16)_t2sint16) : 4096U) : 0U);
    /* Afr_calc: sequence call #2 */
    /* assignment to Afr_facLambdaReq: min=0, max=4096, hex=2048phys+0, limit=(maxBitLength: true, assign: true), zero incl.=true */
    Afr_facLambdaReq__AIRFUEL RATIO _IMPL_Afr_calc
    =
    (_Afr_flgLambdaReqOverride) ? _Afr_facLambdaReqMan : _Afr_facLambdaReqCalc;
    /* Afr_calc: sequence call #3 */
    _tluint32 = (uint32)_Afr_facLambdaReqCalc * _Afr_facStoich;
    /* assignment to Afr_facRatioCalc: min=2048, max=65535, hex=2048phys+0, limit=(maxBitLength: true, assign: true), zero incl.=true */
    _Afr_facRatioCalc = (uint16)(((_tluint32 > 32768U) ? _tluint32 >> 4 : 2048));
    /* Afr_calc: sequence call #4 */
    /* assignment to Afr_facRatio: min=2048, max=65535, hex=2048phys+0, limit=(maxBitLength: true, assign: true), zero incl.=true */
    Afr_facRatio__AIRFUEL RATIO _IMPL_Afr_calc
    =
    (_Afr_flgRatioOverride) ? _Afr_facRatioMan : _Afr_facRatioCalc;
    /* send messages implicitly */
    {

        DisableAllInterrupts();
        Afr_facLambdaReq = Afr_facLambdaReq__AIRFUEL RATIO _IMPL_Afr_calc;
        Afr_facRatio = Afr_facRatio__AIRFUEL RATIO _IMPL_Afr_calc;
        EnableAllInterrupts();
    }
}
/*
-----
* END: DEFINITION OF PROCESS 'AIRFUEL RATIO _IMPL_Afr_calc'
***** */
#endif
/* END: ASCET REGION "Process Definition 'Afr_calc'" */

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

```