

# Function Sheet

Extent shown:  
Blocks chosen: ABK, APP, FB, FDEF, FW

System: SG ME7.2  
Project: BMW 8-Zyl ME-7.2 M62LEV 3,5l und 4,4 l, variable Einl.-NW  
Project code: 11/195;40

Responsible: Markus Riemann  
Department: K3/EAF3  
Phone: 20325  
Date of issue: 19.04.2000

program release: 72111C\_6200 (Predecessor : 72111C\_6100)



## Table of contents: Sections

Page	Section	Version	Descriptor
203	ACIFI	6.10	Output for cylinder-individual injection
548	ADVE	2.12	Activation of the DV-E by means of the DLR
204	AEKP	4.10	EKP control
189	AES	1.50	Overview calculation of injection time
196	AEVAB	2.10	Output injection valve cut off
710	ALE	2.2	Detection of engine stop position
578	AMUENE	1.20	Monitoring counter for maximum engine speed
63	ARMD	2.60	Torque based anti jerk function
7	ASCETBLK	1.0	Description of ASCET block library
10	ASCETSDB	1.23	ASCET-SD description of block library
402	ATEV	2.0	Purge valve drive (duty cycle)
302	ATM	21.40	Exhaust temperature model
759	BBACC	1.30	Operating conditions of ACC (adaptiv cruise control)
475	BBBO	2.10	Start operating range with fuel in oil
492	BBFGR	3.60	Switch-on and switch-off conditions for vehicle-speed controller
70	BBGANG	16.20	Detection of actual gear
282	BBKHZ	17.60	Control of catalyst heating
67	BBKURU	1.10	Operation conditions of slip clutch
746	BBLOR	1.30	Operating gearbox low range
574	BBNMIN	1.1	Lower engine speed limit
579	BBPEDBR	1.20	plausibility check PWG - brake pedal
71	BBSAWE	22.50	Conditions for fuel cut-off / cut-in
575	BBS TT	11.20	Condition Engine start
371	BBTEGA	3.40	Operating conditions for purge canister control / fuel adaption
38	BBVL	1.20	Operating Range Full Load
876	BGCVN	3.10	Calculation value calibration verification Number CVN
534	BGDVE	2.50	Values for DV-E control from the learning and checking routines
861	BGKMST	5.10	Calculation of odometer value (Km)
459	BGKV	1.20	Calculation variable consumed fuel
99	BGLBZ	5.10	Calculated charge deficit of the battery
44	BGMDLM	1.70	Calculated variable: engine torque with torque interventions
69	BGMIGS	1.10	Torque reduction automatic-transmission shift control request
758	BGMRACC	1.10	Calculated variable relative ACC torque
286	BGMSABG	2.10	Calculation of exhaust emission mass flow - bank-dependent
111	BGMSZS	10.60	Calculation of mass flows into intake manifold
565	BGNG	7.10	Calculated variable: engine-speed gradient
578	BGNL DSC	1.10	DSC Extended Power Down Request
564	BGNMOT	4.10	Calculated variable: engine speed
407	BGPUK	1.10	Calculation and correction of value ambient pressure (detection down hill)
703	BGRBS	2.30	Calculated wheel acceleration from wheel speed
125	BGR LP	3.20	Calculation variable rip: predicted air charge
854	BGRML	1.20	Calculation Value of Relative Air Mass according to SAE J1979 Mode \$01 + \$02 PID
118	BGSRM	9.10	Model of intake manif. for calc. relative air charge and intake manif. pressure
154	BGTABST	3.60	Calculated variable: cut-off time
123	BGTEMPK	3.40	Calculation of temperature compensation for intake manifold model
143	BGTEV	1.60	Calculation variable, mass flow TEV
282	BGUBF	1.10	Calculated variable: ub filtered
498	BGVFGR	1.20	Calculated speed for vehicle-speed controller
153	BGWDKM	1.0	Calculation of throttle angle model
752	CANACC	1.100	CAN message and signal definition ACC
732	CANASR	1.80	CAN message and signal definition drive-slip regulation
718	CANDME	1.180	DME CAN message and signal definitions
723	CANECS	1.90	CAN message and signal definition transmission control
727	CANINS	1.110	CAN message and signal definition, combined instrument
731	CANLWS	1.10	CAN message and signal definition steering-wheel angle sensor
742	CANTXU	1.10	CAN message and signal definition Transfer Box Control ECU (TXU)
717	CANUE	1.40	CAN overview; messages and identifiers
203	CIFINA	1.30	CIFI new parts ME7
193	COSA	6.20	Idle CO adjustment
756	DACC	1.60	Plausibility check ACC signal diagnostics
740	DASC	1.40	Diagnostics for ASC signals
755	DCACC	1.30	Diagnosis; CAN timeout ACC interface
739	DCAS	8.30	Diagnosis CAN timeout ASC interface
726	DCGE	13.20	Diagnosis; CAN timeout GE interface
728	DCINS	5.20	Diagnostics; CAN timeout instrument (Combi)
745	DCTXU	1.10	Diagnosis; CAN timeout interface TXU
856	DDCY	14.0	OBDII; fulfillment condition 'driving cycle'
568	DDG	11.10	Diagnosis: engine speed sensor
830	DDMMVE	4.20	Diagnosis of Power Stage of DM-TL Solenoid
828	DDMPME	1.30	Diagnosis of Power Stage of DM-TL Pump Motor
411	DDMTL	1.112	Diagnosis of Tank Leakage with DM-TL Module
813	DECJ	16.11	Diagnosis; Power stage CJ9x
116	DEGFE	2.10	Diagnosis of input variables for charge detection
812	DEKON	3.40	Configuration of power stage diagnosis
824	DEKPE	11.11	Diagnosis; power stage of fuel pump relay
839	DEPCL	1.20	Diagnosis; electronic powertrain control lamp
834	DETS	2.20	Power stage diagnosis; electric thermostat control (OL)
835	DEVE	6.30	Diagnosis; power stage of injector valve
847	DFCM	1.10	OBDII; description fault code memory
849	DFFT	1.40	Diagnostics; Freeze frame selection table
852	DFFTCNV	1.51	Diagnosis; freeze frame table, conversion to bytes



Page	Section	Version	Descriptor
851	DFFTK	2.70	Diagnosis; customer-specific list for selection of freeze-frame values
841	DFPM	3.20	OBDII; Fault path manager
848	DFPMNL	1.20	Diagnosis; OBDII fault path management afterrunning condition
847	DFRZ	20.20	OBDII; description 'freeze frame'
449	DFSTT	1.40	Diagnostics for fluid level sensor tank
747	DGLOR	2.30	Diagnosis; gear in low range
440	DHDMTE	2.10	power stage diagnosis heater DM-TL
108	DHFM	63.20	Diagnosis; plausibility test hot film air flow sensor
666	DHLS	31.10	Diagnosis; Sensor heating
857	DIMC	28.30	OBDII; inspection/maintenance-ready
310	DKAT	59.30	Diagnosis; catalyst conversion
816	DKOSE	6.20	Diagnosis of power stage for AC compressor
247	DKRNT	11.10	Diagnosis; Knock-control, zero-test (OBDII)
242	DKRS	28.10	Diagnosis; knock sensor
252	DKRTP	11.10	Diagnosis of knock control, test pulse for OBDII
561	DKUPPL	1.40	Diagnosis of clutch switch
476	DKVS	17.60	Diagnosis; plausibility test fuel supply system
404	DLDL	16.31	Diagnosis of Tank Leakage with DLDL Module
409	DLDP	4.30	Driver diagnosis; overpressure pump of fuel tank leakage diagnosis
94	DLLR	36.10	Diagnosis: idle speed control, recognising a blocked actuator
634	DLSA	51.10	Lambda sensor aging monitoring
645	DLSAHK	8.10	Ageing monitoring for lambda sensor downstream of catalytic converter
627	DLSH	29.10	Diagnosis; Readiness for operation of sensor downstream catalyst
654	DLSSA	14.80	Signal output from lambda sensors
609	DLSV	36.10	Diagnosis; Readiness for operation of sensor upstream catalyst
625	DLSVV	3.30	diagnosis; detection of exchange lambda sensors upstream catalyst
766	DMDDLU	7.40	Diagnostic routine Misfire Detection, forming the diff. for irregular running
812	DMDFLU	2.10	Diagnostic misfire detection filtering irregular running
772	DMDFON	6.20	Diagnosis Misfire Detection fuel-on Adaptation
799	DMDLAD	3.10	Logic and Delay; Logical operation, different blocks for misfire detection
761	DMDLU	4.60	Diagnostic routine misfire detection: Irregular running
770	DMDLUA	4.10	Diagnostic routine Misfire Detection irregular running spacing
800	DMDMIL	3.30	Fault treatment of misfire detection, control on MIL and rectification
799	DMDSTP	4.60	Diagnostic of misfire detection : Stop conditions
810	DMDTSB	1.20	Diagnosis misfire detection by segment time formation
760	DMDUE	9.90	Diagnostic routine misfire detection Overview
860	DMIL	28.30	OBDII; MIL control
837	DMILE	8.20	OBDII; MIL-power stage
818	DMLSE	7.20	Diagnosis; power stage check of electric cooling fan
267	DNWS	12.30	Diagnosis camshaft control
819	DNWSE	6.20	Diagnosis; camshaft control power stage
571	DPH	20.20	Diagnosis; plausibility test phase sensor
823	DSLPE	9.20	Diagnosis; plausibility check power stage of secondary air pump
289	DSLRLR	2.30	Diagnosis secondary air system with two-step Lambda control
821	DSLVE	13.20	Diagnosis; plausibility check power stage of secondary air valve
709	DSTA	3.10	Diagnosis; automatic start output signals
832	DSTAE	2.20	Diagnosis; powerstage automatic start control
708	DSTS	1.20	Diagnosis; automatic start input signals
704	DSWEC	5.10	Bumpy road detection from wheel acceleration, -> via CAN from ABS CU
826	DTEVE	9.30	Diagnosis; power stage of canister purge valve
403	DTEVN	22.100	Diagnosis of canister purge valve (OBDII)
689	DTHM	4.70	diagnosis: thermostat monitoring
862	DTIP	1.20	OBDII; tester interface package
840	DTOP	3.10	Diagnosis; operating time
854	DTRIG	1.10	OBDII; Selectable trigger for fault path management
673	DUSREF	1.20	Diagnosis: 5 V internal power voltage
932	DUF	4.10	Diagnostic interface of the function monitoring
855	DUMWEX	3.10	Extended ambient conditions
933	DUR	1.20	Diagnosis from computer monitoring
524	DVEUE	1.0	Overview of DV-E-control
581	DVFZ	21.10	Diagnosis; plausibility test vehicle speed
856	DWUC	13.0	OBDII; fulfillment condition 'warm up cycle'
184	EA	139.10	Mode of injection
184	EASTT	3.1	Injection at start
943	EEDAT	2.20	Data in EEPROM
939	EEPROM	11.10	EEPROM treatment
104	EGFE	2.0	Input variables for charging detection
216	EGKE	1.1	Input variables for knocking detection
839	ELS	1.10	Fan control ecu box
161	ESGRU	32.10	Basic function injection
172	ESKS	1.20	Knock protection injection
166	ESNST	3.90	Afterstart enrichment
174	ESNWS	1.0	Camshaft control for mixture correction
162	ESSTT	16.30	Injection duration at start
175	ESUK	2.60	Injection: transient compensation
160	ESVST	5.10	Fuel injection pre-control
174	ESWE	1.10	Injection, resumption of overrun fuel cut-off
171	ESWL	7.50	Injection for warm-up
688	ETR	8.20	Electric thermostat control
511	FGRAUS	1.0	Cut-out by vehicle-speed controller
499	FGRMD	4.10	Function logic and control algorithm of vehicle-speed controller
490	FGRUE	2.10	Overview of vehicle-speed controller



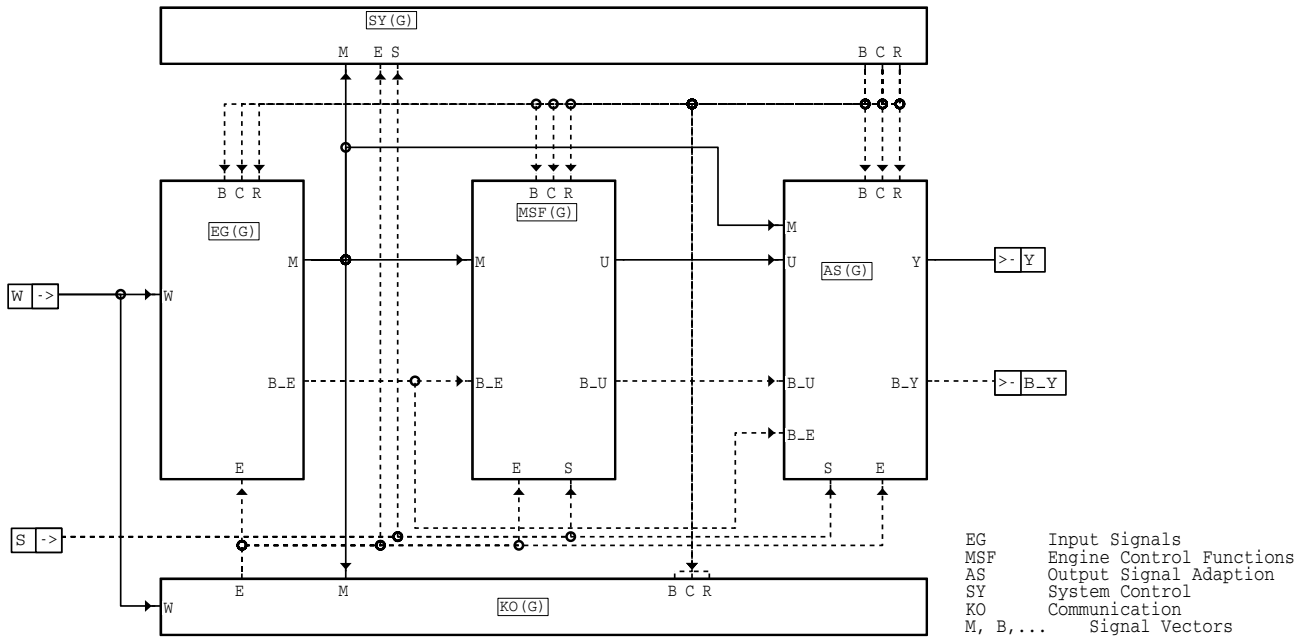
Page	Section	Version	Descriptor
149	FUEDK	6.20	Charge control (calculation of nominal throttle-valve angle)
730	GGAIRB	1.20	Sensor variable: airbag
562	GGDG	1.5	Pickup variables: speed pickup and reference mark pickup
410	GGDML	3.10	Signal DM-TL pump current output
131	GGDSAS	4.50	Sensor variable for pressure sensors outside the manifold pipe
525	GGDVE	1.51	Sensor variables for throttle-valve actuator
560	GGEVAS	8.20	Sensor variable for brake and clutch switches
443	GGFSTT	1.40	Sensor variable fluid level sensor tank
105	GGHFM	53.50	Sensor signal, hot-film air-mass meter
218	GGKS	7.10	Sensor signal for knocking detection
599	GGLSH	3.30	Sensor variable for lambda sensor downstream of catalytic converter
588	GGLSV	2.30	Sensorsignal lambda upstream catalyst
490	GGMFL	1.70	Sensor variable for multi-function steering wheel
513	GGPED	1.50	Sensor variable for accelerator pedal
523	GGPEDV	1.10	conditioning input signals of accelerator pedal
681	GGTFA	15.10	Diagnosis; intake air temperature sensor
674	GGTFM	36.40	Signal engine temperature sensor
684	GGTFUMG	4.40	Signal ambient temperature sensor
683	GGTKA	1.10	Sensor variable for radiator-outlet temperature, TKA
686	GGTSG	1.10	Sensor variable and documentation of temperature control unit
671	GGUB	11.10	Sensor variable for battery voltage incl. diagnosis
580	GGVFZG	10.20	Input signal: vehicle speed
158	GK	3.40	Overview, mixture control
325	GKEB	3.0	Operating condition mixture control overview
327	GKRA	3.0	Mixture control and adaptive pilot control
442	HDML	1.20	Heating DM-TL
582	HLS	16.20	Lambda sensor heater
324	KATV	1.20	Variant recognition (vehicle with/without cat. converter and lambda sensors)
287	KHMD	2.20	Calculation of torque reserve for heating catalytic converter
698	KOS	107.10	Control of A/C compressor
237	KRDY	19.10	Knock control for load dynamics
219	KRKE	10.12	Knocking detection
227	KRRA	16.80	Adaptive knock control
713	KVA	42.20	Output signal: display of fuel consumption
272	LAKH	3.20	Lambda coordination for catalyst heating
271	LAMFAW	3.30	Lambda vehicle-operator demand
268	LAMKO	3.140	Lambda coordination
100	LLRABG	1.20	Idle-speed control - correction of nominal speed by tester
81	LLRBB	2.40	Operating conditions of idle speed control
83	LLRMD	1.2	Torque-based idle-speed control
92	LLRMR	4.40	Torque reserve for idle speed control
84	LLRNS	517.30	Idle control; Nominal engine speed for idle speed control
88	LLRRM	7.20	Idle speed control: torque controller
338	LR	74.50	Lambda closed loop control
463	LRA	117.10	Lambda closed loop control; Adaptive pilot control
461	LRAEB	5.40	Conditions adaptive pilot control
328	LREB	123.60	Activation conditions for lambda closed loop control
365	LRHK	33.10	Lambda closed loop control downstream catalyst (OBDII)
352	LRINI	1.0	Coordination initializing lambda controller
353	LRKA	15.10	Two Sensor Lambda Control: Oxygen Clear Out Function
50	MDBAS	4.30	Basic calculation for torque interface
31	MDFAW	3.150	Calculation of vehicle-operator demand
148	MDFUE	9.20	Nominal-value input from nominal torque for airmass
68	MDIHGS	1.60	Torque limiting as a function of the gear
49	MDIST	10.2	Engine torque calculation
40	MDKOG	9.50	Coordination torque intervention
48	MDKOL	3.10	Coordination torque intervention air path
39	MDMAX	4.20	Calculation maximum torque
51	MDMIN	5.10	Minimum engine torque coordination
59	MDNSTAB	1.10	Torque: engine-speed stabilization
73	MDRED	11.10	Calculation reduction step from torque demand
301	MDTRIP	2.10	Calculation of torque reserve for short trip
53	MDVER	1.30	Loss in engine torque
55	MDVERAD	5.40	Adaptation of torque loss
52	MDVERB	2.0	Torque demand by auxiliary systems (e.g. air conditioner, misc. consumers)
61	MDWAN	5.10	Torque of the AT-converter
45	MDZUL	4.10	Calculation of maximum permitted set torque
211	MDZW	1.90	Calculation of torque in nominal ignition timing
695	MLS	49.50	Electric cooling fan control
576	MOTAUS	12.10	Engine switch-off
936	MOTOR	245.10	Engine data
6	MS	3.0	Engine control overview
30	MSF	9.10	Engine control functions
80	MSMD	1.10	engine protection through torque reduction
76	NMAXMD	5.30	Torque calculation during maximum speed control
862	OBDSV	1.20	OBD status management
878	PROKON	8.110	Project configurations
192	RKTI	1.60	Calculation of injection time $t_i$ from relative fuel mass $r_k$
214	RUV	42.0	Distributorless ignition
863	SCATT	20.30	SCAN TOOL-tester interface
275	SLS	91.20	Secondary air control
931	SREAKT	5.10	EGAS: safety concept, failure reactions



Page	Section	Version	Descriptor
706	STA	7.20	Automatic start control
715	STECK	394.30	Plug pin arrangement
60	STMD	4.10	Starting torque
28	SYABK	8.2	Symbols and abbreviations
939	T2DCLA	1.10	KWP2000; reset adaptive value
936	T2DFA	1.40	KWP2000; activation of diagnostic function
941	T2DMTL	1.40	KWP2000: Request service DMTL system test
864	TC1MOD	20.80	Tester communication CARB; Mode 1
868	TC2MOD	21.20	Tester communication CARB; Mode 2
870	TC5MOD	20.10	Tester communication CARB; Mode 5
871	TC6MOD	21.50	Tester communication CARB; Mode 6
872	TC8MOD	20.0	Tester communication CARB; Mode 8
872	TC9MOD	7.10	Tester communication CARB; Mode 9, Request vehicle information
876	TCSORT	3.10	Tester communication CARB; sort function
380	TEB	95.10	Purge canister function
378	TEBEB	9.10	Switch-on conditions for purge control
687	UETM	2.10	Engine overtemperature display
891	UFEING	10.10	ETS monitoring concept: Input signal transfer used in function monitoring
900	UFFGRC	2.50	ETS monitoring concept: Monitoring of Cruise Control of function monitoring
902	UFFGRE	2.30	ETS monitoring concept: CC input information used in function monitoring
897	UFMIST	2.10	ETS monitoring concept: Calculation of the actual torque in UF
903	UFMSRC	6.30	ETS monitoring concept: MSR intervention surveillance for function monitoring
905	UFMVER	2.10	ETS monitoring concept: Torque comparison of function monitoring
893	UFMZP	1.10	ETS monitoring concept: Torque filter for function monitoring
894	UFMZUL	5.10	ETS monitoring concept: Calculation of permissible torque in UF
906	UFNC	2.20	ETS monitoring concept: Monitoring of engine speed for function monitoring
896	UFNSC	2.10	ETS monitoring concept: Afterstart monitoring for function monitoring
909	UFOBP	1.10	ETS monitoring concept: OBP operation of function monitoring
898	UFREAC	2.20	ETS monitoring concept: Monitoring of fault reaction of function monitoring
909	UFRLC	4.10	ETS monitoring concept: Monitoring of load signal for function monitoring
913	UFSPSC	3.11	ETS monitoring concept: Monitoring of accelerator pedal value for function m.
890	UFUE	4.10	ETS monitoring concept: function monitoring overview
913	UFZWC	2.20	ETS monitoring concept: Monitoring of ignition angle for function monitoring
915	UMAUSC	1.11	ETS monitoring concept: test of the shut-down path of the monitoring module
916	UMFPW	2.10	ETS monitoring concept: flash programming request in the monitoring module (UM)
917	UMFSEL	1.10	ETS monitoring concept: inquiry selection in the monitoring module(UM)
918	UMKOM	2.10	ETS monitoring concept: Inquiry/response communication between UM/FR
921	UMTOUT	2.10	ETS monitoring concept: time-out for UM/FR-communication
923	URADCC	2.20	ETS monitoring concept: test of the AD-converter
924	URCPU	2.10	ETS monitoring concept: Instruction test by means of level 2'
926	URMEM	1.11	ETS monitoring concept: cyclic memory test
928	URPAK	1.10	ETS monitoring concept: program flow check
929	URRAM	2.10	ETS monitoring concept: RAM-test
930	URROM	2.20	ETS monitoring concept: ROM-test
78	VMAXMD	3.30	Torque request of Vmax regulation
714	VS_VERST	2.40	Adjusting parameters for McMess
257	WANWKW	11.50	Angle adaptation of alignment between camshaft and crankshaft
254	WNWEIN	5.30	Nominal camshaft angle for intake camshaft position regulation
261	WNWR	4.40	Camshaft position control with adaptations
267	WNWTST	1.10	system diagnosis request of camshaft controller
215	ZNDAUS	8.10	Ignition: requirements for suppression
206	ZUE	258.100	Basic function - ignition
207	ZUESZ	2.10	Ignition, calculation of coil closing time
208	ZWGRU	18.20	Basic ignition angle
212	ZWKS	1.10	Ignition timing knock protection
213	ZWMIN	1.50	Calculation of maximum retarded spark limitation
210	ZWSTT	4.10	Ignition at start
210	ZWWL	4.10	Ignition during warm-up

## MS 3.0 Engine control overview

### FDEF MS 3.0 Function definition



ms-ms

### ABK MS 3.0 Abbreviations

Variable	Source	Type	Description
B	MS	LOK	vector operating state conditions (only CU model)
B_E	MS	LOK	bit vector input variables
B_U	MS	LOK	bit vector manipulated variables
B_Y	MS	AUS	bit vector output variables
C	MS	LOK	vector controller state conditions (only CU model)
E	MS	LOK	Vector of error-flags
M	MS	LOK	vector engine variables physical (only CU model)
R	MS	LOK	vector computing base flags (only CU model)
S		EIN	vector switch flags (only CU model)
U	MS	LOK	revolution counter
W		EIN	vector converted input variables (only CU model)
Y	MS	AUS	vector of physical CU output variables

### FB MS 3.0 Detailed description of function

The control unit model of the Motronic is made up of the following main parts:

- EG Input variables: Processing and diagnosis of the sensor signals, derivation of further variables
- SY System variables: Initializations, calculation cycles etc.
- MSF Engine control functions: Contain the functional groups to calculate the setting variables, e.g. torque coordination, injection, ignition, idle control etc.
- AS Output signals: Conversion of the setting variables into hardware-dependent setting signals
- KO Communication: Representation of the interfaces to the fault code memory management, customer service, driver, application etc.

The structure of the SG functions has been revised. The aim is a modular and more comprehensible structure. Checking and data flows are separated to a great extent. Data and checking flows are represented by different lines. Transfer of the data on the upper structure levels is performed by vectors.

## APP MS 3.0 Application hint

## ASCETBLK 1.0 Description of ASCET block library

### FDEF ASCETBLK 1.0 Function definition

#### Funktionsdarstellung:

Bei der Darstellung von Funktionen wird zwischen physikalischen Informationen (Datenfluß) und digitaler Steuerinformation (Kontrollfluß) unterschieden.

Datenfluß: Lastsignal, Drehzahl, Regelfaktor

Kontrollfluß: Bedingung Leerlauf, Schalter Fahrstufe, Fehler Kat

Durchgezogene Linien markieren den Datenfluß, gestrichelte Linien den Kontrollfluß.



#### Grundblöcke (allgemeines):

- Bei Blöcken mit der Kennzeichnung "NOV" am Ausgang wird der Zustandswert des Blockes (Integratorinhalt, Flag, RAM-Zelle, etc.) im Dauer-RAM gespeichert (ansonsten im flüchtigen RAM). Im übrigen verhalten sich die Blöcke wie ihre Pendants ohne "NOV".
- Die Haupteingangs- und Hauptausgangswerte ("in" und "out") weisen im Block-Icon kein Symbol auf; sie sind mit 0.0 (float) bzw. FALSE (bool) vorbelegt, sofern nichts anderes angegeben ist.
- Nichtbeschaltete Eingänge sind mit 0.0 (float) bzw. FALSE (bool) vorbelegt, sofern nichts anderes angegeben ist.
- Bei einigen Blöcken kann an der linken oberen Ecke ein "Rastereingang" (default TRUE) angeschlossen werden, durch den die Berechnungshäufigkeit explizit festgelegt wird. Im folgenden bezeichnet "rasterZeit" den Abstand zwischen zwei Berechnungen.
- Eine Abweichung von der nachfolgenden Standardbelegung der Ein- und Ausgängen wird in der Beschreibung des Blockes angegeben.

EINGÄNGE:	Kürzel im Icon	Default-Wert	Bezeichnung
E	E	TRUE	Berechnung des Blocks freigeben
I	I	FALSE	Initialisierung auslösen
IV	IV	0.0	Initialisierungswert
K	K	0.0	hier: Integrationsfaktor K
MX	MX	1E35	obere Begrenzung der Ausgangsgröße
MN	MN	-1E35	untere Begrenzung der Ausgangsgröße

#### ascetblk-teil0



#### Integrator K

neuer Integratorwert := alter Integratorwert + K \* rasterZeit \* in

EINGÄNGE: K Integrationsfaktor



#### Integrator T

neuer Integratorwert := alter Integratorwert + (rasterZeit / T) \* in

Der Minimalwert von T wird auf rasterZeit begrenzt.

EINGÄNGE: T Integrationszeitkonstante



#### Rekursion

neuer Wert := alter Wert + m \* (in - alter Wert)

EINGÄNGE: m Rekursionskonstante



#### Tiefpass

neuer Tiefpasswert := alter Tiefpasswert + (rasterZeit / T) \* (in - alter Tiefpasswert)

Der Minimalwert von T wird auf rasterZeit begrenzt.

EINGÄNGE: T Zeitkonstante



#### Eingangs-UmschalterUnten

Das Icon zeigt die Ruhestellung des Schalters, nichtbeschaltete Eingänge sind mit 0.0 vorbelegt.



#### Exklusiv-ODER

Der Ausgang wird TRUE, wenn genau ein Eingang TRUE ist.



#### FlankeBi

Bei negativer oder positiver Flanke am Eingang, wird während dieses Simulationsschrittes am Ausgang TRUE ausgegeben. Sonst ist der Ausgang FALSE.



#### Maximum2

Am Ausgang liegt das Maximum der Eingangswerte an.

Der Ausgang i zeigt den Index des ersten Eingangs an, dessen Wert gleich dem ermittelten Maximum ist.

#### ascetblk-teil1



**Begrenzer**

Am Ausgang wird der auf den Bereich [MN, MX] begrenzte Eingangswert ausgegeben.  
Ist eine Begrenzung aktiv, so wird der Ausgang B := TRUE gesetzt; ansonsten ist dieser Ausgang FALSE.



**Betrag**

Am Ausgang liegt der Betrag des Eingangswertes an.



**Hystrese**

Der rechte und der linke Schaltppunkt der Hysteresis ergibt sich aus der Beschaltung:  
beschaltet linker Schaltppkt rechter Schaltppkt

LSP und delta	LSP	LSP + delta
LSP und RSP	LSP	RSP
delta und RSP	RSP - delta	RSP

Bei allen anderen Beschaltungen der Eingänge wird am Ausgang FALSE ausgegeben (fehlerhafte Beschaltung).



**Signum**

Ist der Eingangswert < 0.0, liegt am Ausgang der Wert -1.0, ansonsten der Wert 1.0 .



**Akkumulator**

Der Akkumulator wird um den Eingangswert additiv verändert und auf den Bereich [MN, MX] begrenzt.



**FLAG**

Nachbildung einer flüchtigen 1 Bit-Speicherzelle.



**RAM**

Nachbildung einer flüchtigen Speicherzelle.

**ascetblk-teil2**



**RS-FlipFlop**

Das RS-FlipFlop hat einen Set-Eingang S und einen Reset-Eingang R.  
Am Ausgang !Q liegt immer der zu Q invertierte Wert. Reset ist gegenüber Set dominant.



**VerzögerungRaster**

Verzögerung des Signals um ein Raster, d.h. out(i) := in(i-1).  
Am Ausgang liegt der jeweils um einen Rastertakt verzögerte Wert an.  
Wenn der Rastereingang offen ist, wird um einen Simulationstakt verzögert.



**Ausschalt-Verzögerung**

Der Ausgang folgt dem Schalten des Eingangs von TRUE nach FALSE nach der Verzögerungszeit, die am Eingang DELAY anliegt. Schaltet während der Verzögerung der Eingang wieder nach TRUE, liegt auch am Ausgang sofort TRUE an.



**Einschalt-Verzögerung**

Der Ausgang folgt dem Schalten des Eingangs von FALSE nach TRUE nach der Verzögerungszeit, die am Eingang DELAY anliegt. Schaltet während der Verzögerung der Eingang wieder nach FALSE, liegt auch am Ausgang sofort FALSE an.



**Timer**

Eine positive Flanke am Eingang bewirkt, daß der Timer gestartet wird, d.h.  
- der interne Timer wird auf den Wert (in Sekunden) gesetzt, der am Eingang SV anliegt,  
- der Ausgang wird TRUE und bleibt TRUE bis der Timer abgelaufen ist.  
Eine erneute positive Taktflanke am Eingang hat keine Auswirkung, solange der Timer noch nicht abgelaufen ist. Liegt an E FALSE, wird der Timer gestoppt, bis E wieder TRUE ist.

EINGÄNGE: in Starten des Timers  
SV Timerzeit  
AUSGÄNGE: B Timer läuft



**Timer-Retrigger**

Grundfunktion wie "Timer", jedoch: Eine erneute positive Taktflanke am Eingang bewirkt stets Neustart des Timers.

**ascetblk-teil3**





**Zeitzähler**

TRUE am Eingang R setzt den Zeitzähler auf 0.0 zurück. Wird R = FALSE, beginnt der Zeitzähler zu laufen. Liegt an E FALSE, so wird der Zeitzähler gestoppt. Der Zeitzähler zeigt die abgelaufene Zeit in Sekunden an.  
 EINGÄNGE: R Rücksetzen des Zeitzählers



**Zähler**

Dieser Block zählt in jedem Simulationsschritt um eins aufwärts bzw. abwärts. Startwert, Endwert und damit die Zählrichtung werden festgelegt, wenn am Eingang I TRUE anliegt. Wenn der Wert von SV größer als der Wert von EV ist, dann wird abwärts (ansonsten aufwärts) gezählt, bis der Endwert erreicht ist. Das Erreichen des Endwertes wird durch ein TRUE am Ausgang B angezeigt. Der Zähler kann mit dem Eingang E gestoppt werden.  
 EINGÄNGE: SV Startwert des Zählers  
 EV Endwert des Zählers  
 I Zähler starten  
 AUSGÄNGE: B Endwert erreicht



**Zustandsautomat**

Der Kontrollfluß wird durch logische Gatter und Zustandsautomaten dargestellt. In Zustandsautomaten wird der Funktionsablauf in graphischer Form mit Hilfe von "Zuständen" und "Übergängen" abgebildet. Zustand: Innerhalb eines Zustandsautomaten ist jeweils genau ein Zustand aktiv, d.h. die zu diesem Zustand (Ellipse) gehörenden Aktionen werden ausgeführt. Der Name des Zustandes ist innerhalb der Ellipse dargestellt.

Übergang: Der Übergang von einem Zustand zum anderen erfolgt, wenn die Übergangsbedingung erfüllt ist. (Pfeil) Dabei werden diesem Übergang zugeordnete Aktionen ausgeführt. Die Bedingung, die erfüllt sein muß, damit ein Übergang stattfindet, steht neben dem jeweiligen Pfeil; ggf. steht nur ein logischer Name für die Bedingung und die ausführliche Beschreibung ist dem nachfolgenden Text zu entnehmen. Bevorzugt wird die Bedingung mit der niedrigsten Nummer. Für jeden Zustandsautomaten ist festgelegt, welcher Zustand beim Start des Automaten angenommen werden soll (S) und welcher Zustand bei erfüllter RESET-Bedingung (R).

ascetblk-teil4

**ABK ASCETBLK 1.0 Abbreviations**

Variable	Source	Type	Description
N		EIN	Engine speed

ascetblk-teil4

## FB ASCETBLK 1.0 Detailed description of function

### APP ASCETBLK 1.0 Application hint

## ASCETSDB 1.23 ASCET-SD description of block library

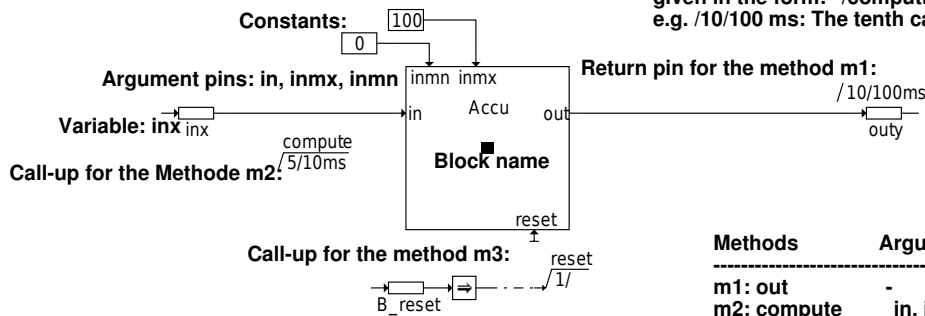
### FDEF ASCETSDB 1.23 Function definition

#### Graphical presentation of basic elements

Basic elements are presented in the diagram as rectangular blocks. Communication between basic elements is displayed by connecting lines. The interfaces between basic elements are the pins at the block edges. Each block has return pin that outputs the result from the block. In addition to this, there are argument pins that provide the inputs into the block as well as method pins that are used for those methods without input arguments and without return values.

The methods call up functions in the block.

The process information and the computing sequence are given in the form: "/computing sequence/process", e.g. /10/100 ms: The tenth call-up in the 10-ms computing frame



Methods	Arguments	Return value
m1: out	-	Float
m2: compute	in, inmx, inmn	-
m3: reset	-	-

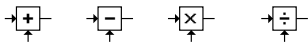
The example given above shows a block with 3 methods:

- The method m1 "out" has one return value
  - The method "out" is called up by the request for the return value from the subsequent block outy, that is in the tenth position in the computing sequence in the 100-ms computing frame.
  - The method m2 "compute" has three arguments (in, inmx, inmn) yet no return value.
  - The method "compute" is called up at the fifth position in the computing sequence in the 10-ms computing frame.
  - The method m3 "reset" has neither arguments nor a return value. This is therefore represented by the "method pin".
- If B\_reset is true, then the method "reset" is called up first (1/) in the computing sequence.

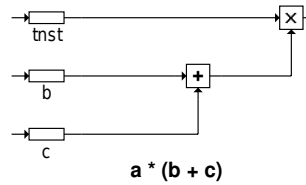
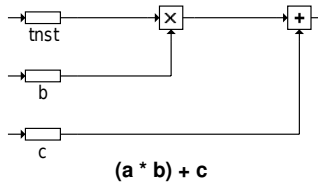
ascetsdb-e-beschrei

ascetsdb-e-beschrei

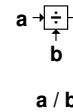
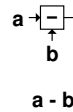
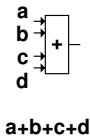
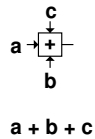
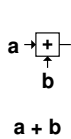
## Arithmetic operations



Equations can be described using arithmetic operations (addition, subtraction, multiplication and division). Equations are represented graphically such that the return value of an operation is the argument for the subsequent operation.



The arguments of primitive operations and their computing sequence are shown in the following:



$a \rightarrow \text{+} \leftarrow b$  **Negation:**  $b = -a$

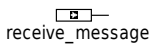
$a \rightarrow \text{|X|} \leftarrow b$  **Amount:**  $b = |a|$

$a \rightarrow \text{MX} \leftarrow b$  **Maximum for input values:**  $c = \text{MAX}(a,b)$

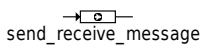
$a \rightarrow \text{MN} \leftarrow b$  **Minimum for input values:**  $c = \text{MIN}(a,b)$

ascetsdb-e1-artihme

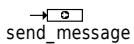
## Variables



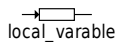
Receive message are input variables of the function that are made available from another function.



Send/Receive variables are output variables of the function that are used both within as well as outside of the function.



Send messages are output variables of the function and are available for the other functions.

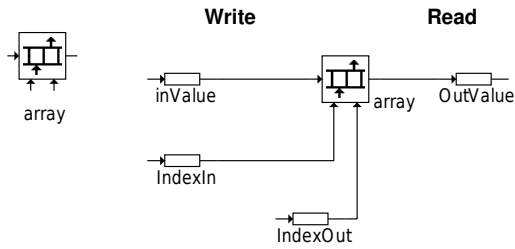


Local variables are only made available and used within the function.

ascetsdb-e2-variabl

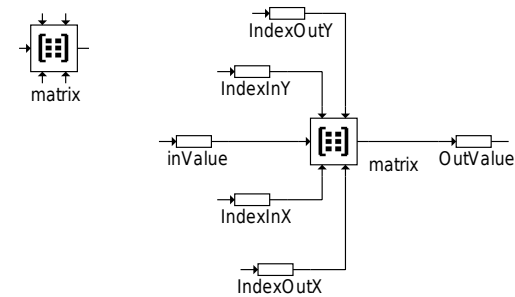
## Arrays and matrices

Arrays and matrices have two methods for reading and writing access to the elements. The reading and writing can be independent of one another.



### Array:

- The value to be written is connected to the left pin, the associated index to the lower left pin.
- The value to be read is connected to the right pin, the associated index to the lower right pin.



### Matrix:

Matrices behave as arrays, whereby the methods here have two index arguments (x, y):

- The index x is connected to the lower left, the index y to the upper left pin for writing access.
- The index x is connected to the lower right, the index y to the upper right pin for reading access.

### ascetsdb-e3-arrays-

### Constants

### Boolean Constants

### ascetsdb-e4-konstan

### System constants

SY\_ZYLZA

System constants are constants that are permanently anchored in the program and cannot be applied. System constants can switch functional positions conditionally on or off.

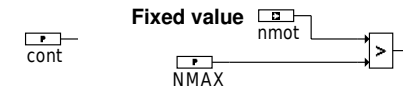
SY\_TURBO

### Example

SY\_ZYLZA: Cylinder number  
SY\_TURBO: Engine with or without turbo-charger

### ascetsdb-e5-systemk

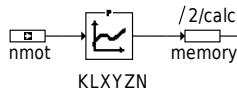
## Fixed values, Characteristics, Maps, Group characteristics, Group maps and datapoint distribution



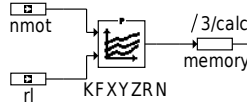
Fixed values are applicable parameters



### Characteristic



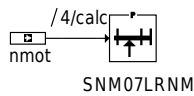
### Map



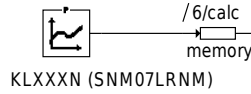
Characteristics have one argument.  
Maps have two arguments as the input.  
Both have one return value



### Datapoint distribution



### Group characteristic



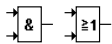
In the case of group characteristics and group maps, several characteristics and maps pick up the same datapoint distributions. To this end, the current datapoint from the datapoint distribution, e.g. SNM07LRNM, is first computed from the dependent parameter e.g. nmot. The computation of the output value for the group characteristic or map then follows using this current datapoint.

### Group map



ascetsdb-e6-kl-kf-g

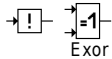
## Bit operations



E1	E2	A
0	0	0
1	0	0
0	1	0
1	1	1



E1	E2	A
0	0	0
1	0	1
0	1	1
1	1	1



E	A
0	1
1	0



E1	E2	A
0	0	0
1	0	1
0	1	1
1	1	0

ascetsdb-e7-bitoper

## Comparator

The comparator provides TRUE at the output if the comparison applies. If the comparison is not fulfilled, then FALSE is given as the output.



Greater, greater than or equal to

The comparison is always read from top to bottom (interval excepted):

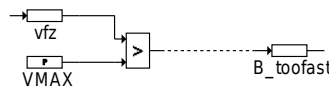


Less, less than or equal to

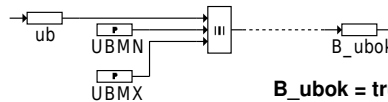
If vFz is greater than VMAX, then the condition B\_toofast is TRUE



Equal, unequal



Closed interval:  
 $a \leq x \leq b$



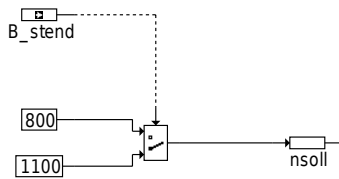
$B\_ubok = \text{true if } UBMN \leq ub \leq UBMX$

ascetsdb-e8-verglei

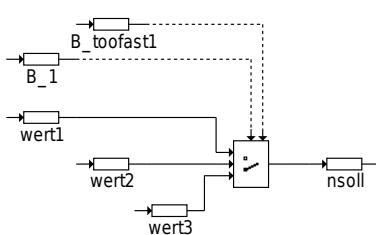


## Multiplex operator "Multiplexer", "Switch"

A multiplexer switches a value through to the output as a dependency of the input conditions. The multiplexer icon is shown in the dormant position, i.e. if the input conditions are false.



**Example "Simple multiplexer"**  
- if B\_stend = false: nsoll = 1100  
- if B\_stend = true: nsoll = 800

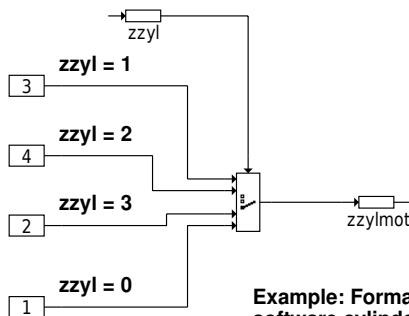


An input condition is assigned to each value in the case of cascaded multiplexers. The uppermost value, the input condition for which is true, is switched through. If there is no input condition of true, then the lowest value is switched through.

**Example "Mehrfach-Muxer":**  
- if B\_1 = true: nsoll = value1  
- if B\_1 = false & B\_2 = true: nsoll = value2  
- if B\_1 = false & B\_2 = false: nsoll = value3

ascetsdb-e9-multipl

## CASE operator



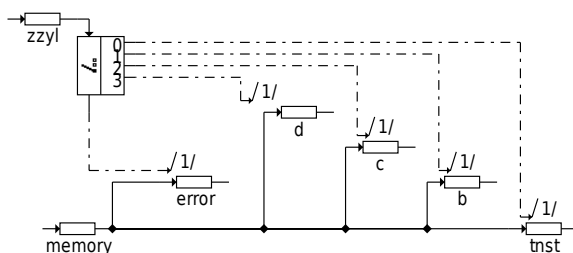
The CASE operator switches one of the remaining left inputs through to the output as a dependency of a top-applied discrete control value (1, 2, 3, ...). If the control value 1 is the first value, then 2 is the second value and is switched through immediately. If the value is outside of the range, then the lowest input (default) is switched through.

**Example: Formation of the physical cylinder number onto the software cylinder number.**

ascetsdb-e10-case-o

## Switch

The SWITCH operator activates the matching control flows over the right-hand outputs as a dependency of an upper applied discrete control value (1, 2, 3, ...). If there is no matching output existing, then the control flow is activated at the lower output.



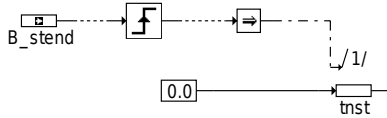
**Example:**  
One of the following operations is executed depending on zzy1:  
- if zzy1 = 0: a = memory  
- if zzy1 = 1: b = memory  
- if zzy1 = 2: c = memory  
- if zzy1 = 3: d = memory  
- otherwise: error = memory

ascetsdb-e11-kontro

### If ..... then



The If .. Then operation analyzes a logic condition and activates all computing sequences for TRUE that are connected to the logic flow. The computing sequence is defined by the numbering.



**Example:**  
If B\_stend changes to true, then tnst = 0 is set.

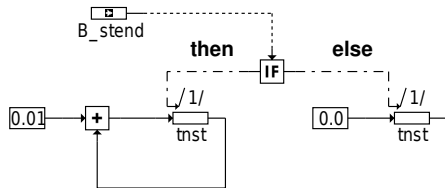
### If .. Then .. Else

If



else

The IF .. Then .. Else operation analyzes a logic condition and activates all computing sequences of the then control branch for TRUE, and all computing sequences of the else control branch for FALSE. The computing sequence at each control branch is defined by the numbering.



**Example:** If B\_stend = true, then tnst is incremented by 0.01 sec. in the 10-ms time frame. Otherwise (B\_stend = false) tnst = 0 is set.

**Mule bridge for IF:**

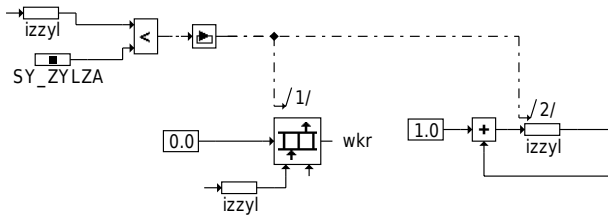
I=True <-- IF --> F=False

ascetsdb-e12-kontro

### While loop



The computing sequence within the control flow of the loop is executed for such a time as the input condition is fulfilled, i.e. is TRUE. The loop is aborted when the input condition is FALSE. The value for terminating the While loop is normally formed within the loop. This usually concerns a counter here that shall count up to a certain value.



**Example:**  
The array wkr[i] is written with 0 for such a time as izzy < SY\_ZYLZA. Each element of the array is initialized with 0 by the numerical variable izzy at the index input of the array.

ascetsdb-e13-while

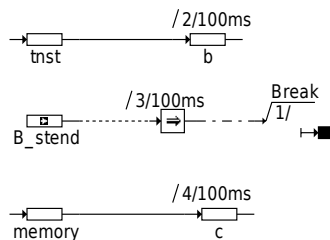
### Break

Break



The break operation prematurely interrupts a process, e.g. the functions component in a computing frame. All subsequent calculations of the function in the process with a higher number for the computing sequence are not then executed.

**Example:**



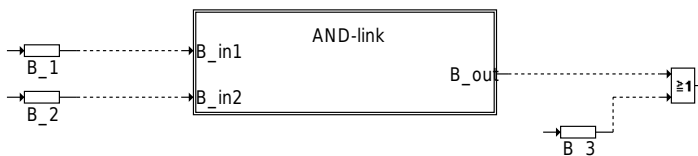
In accordance with the computing sequence, a break is triggered after the operation b = a exactly then when B\_stend = TRUE. The 100-ms process is started if a break occurs. The subsequent operation, c = memory, is no longer executed.

ascetsdb-e14-kontro

### Hierarchy:



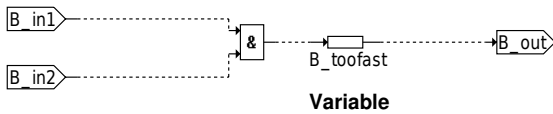
The hierarchy is graphical possibility for structuring functions. The hierarchy block is identified by a double border. The corresponding hierarchy level is identified by the name, here "AND link". The transfer element is only a designator for the links between both levels.



ascetsdb-e15-hiarar

## Function in the graphical "AND link" hierarchy:

### Designator for input links

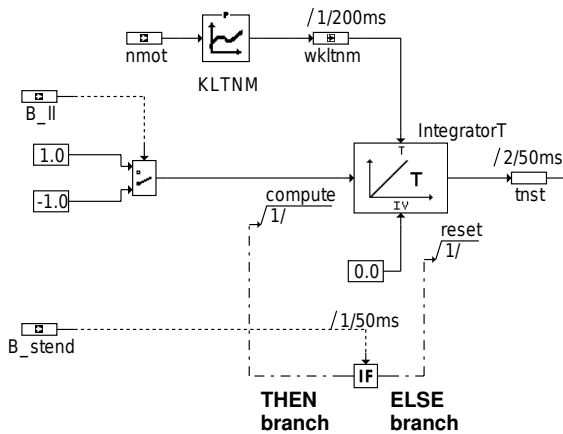


Variable

### Designator for return links

ascetsdb-and-link

### Example

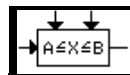


The method "reset" of the integrator is executed in the 50-ms computing frame as long as the condition  $B\_stend = FALSE$ . This method causes the internal memory cell of the integrator to be initialized with the IV-value, i.e. 0.0. If the condition now becomes  $B\_stend = TRUE$ , then the left-hand control flow is activated and the method "compute" has as its arguments time T and the input value. This depends on  $B\_II$ . The input value = 1.0 for  $B\_II = TRUE$ ; the input value = -1.0 for  $B\_II = FALSE$ . The number in front of the computing frame indicates the computing sequence: The time constant T is computed in the 200-ms computing frame and the message wkltnm is stored in Send/Receive. The IF .. THEN .. ELSE query is first executed in the 50-ms computing frame. The integration value is written into the variable a in the second step.

ascetsdb-e-b-beispi

## ASCET-SD System Library

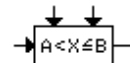
### Comparators



#### ClosedInterval

ClosedInterval returns TRUE, if the value x is in the closed interval defined by A and B.

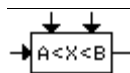
Methods	Characteristics	Arguments	Return Value
out	TRUE is returned, if $A \leq x \leq B$ . Otherwise FALSE is returned.	x::continuous A::continuous B::continuous	TRUE or FALSE



#### LeftOpenInterval

LeftOpenInterval returns TRUE, if the value x is in the left open interval defined by A and B.

Methods	Characteristics	Arguments	Return Value
out	TRUE is returned $A < x \leq B$ . Otherwise FALSE is returned.	x::continuous A::continuous B::continuous	TRUE or FALSE



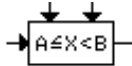
#### OpenInterval

OpenInterval returns TRUE, if the value x is in the open interval defined by A and B.

Methods	Characteristics	Arguments	Return Value
out	TRUE is returned, if $A < x < B$ . Otherwise FALSE is returned.	x::continuous A::continuous B::continuous	TRUE or FALSE

ascetsdb-etassys1





### RightOpenInterval

RightOpenInterval returns TRUE, if the value x is in the right open interval defined by A and B.

Methods	Characteristics	Arguments	Return Value
out	TRUE is returned, if $A \leq x < B$ . Otherwise FALSE is returned.	x::continuous A::continuous B::continuous	TRUE or FALSE



### GreaterZero

GreaterZero returns TRUE, if the value x is greater than zero.

Methods	Characteristics	Arguments	Return Value
out	TRUE is returned, if $x > 0.0$ . otherwise FALSE is returned.	x:continuous	TRUE or FALSE

## Counter & Timer

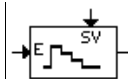


### CountDown

CountDown decrements the counter and signals when the counter has reached zero.

Methods	Characteristics	Arguments	Return Value
start	The counter is set to the start value .	startValue::unsigned discrete	none
compute	The counter is decremented by one.	none	none
out	TRUE is returned if the counter is greater than zero. Otherwise FALSE is returned.	none	TRUE or FALSE

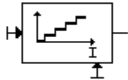
### ascetsdb-etassyse2



### CountDownEnabled

CountDownEnabled decrements the counter and signals when the counter has reached zero. This counter must be enabled explicitly.

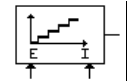
Methods	Characteristics	Arguments	Return Value
start	The counter is set to start value.	startValue:: unsigned discrete	none
compute	If enable is TRUE, the counter is decremented by one.	enable::TRUE or FALSE	none
out	TRUE is returned if the counter is greater zero. Otherwise FALSE is returned.	none	TRUE or FALSE



### Counter

Counter increments the counter by one.

Methods	Characteristics	Arguments	Return Value
reset	The counter is set to zero.	none	none
compute	The counter is incremented by one .	none	none
out	The counter value is returned .	none	unsigned discrete

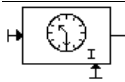


### CounterEnabled

CounterEnabled increments the counter by one. This counter must be enabled explicitly.

Methods	Characteristics	Arguments	Return Value
reset	If <code>initEnable</code> is TRUE, the counter is set to zero .	initEnable::TRUE or FALSE	none
compute	If <code>enable</code> is TRUE, the counter is incremented by one.	enable::TRUE or FALSE	none
out	The counter value is returned.	none	unsigned discrete

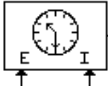
### ascetsdb-etassyse3



### StopWatch

StopWatch increments the time counter by one dT.

Methods	Characteristics	Arguments	Return Value
reset	The counter is set to zero.	none	none
compute	The time counter is incremented by dT.	none	none
out	The time counter value, i. e. the time elapsed since the last start, is returned.	none	continuous

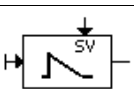


### StopWatchEnabled

StopWatchEnabled increments the time counter by one dT. This timer must be enabled explicitly.

Methods	Characteristics	Arguments	Return Value
reset	If <code>initEnable</code> is TRUE, the internal counter is set to zero.	<code>initEnable::TRUE</code> or FALSE	none
compute	If <code>enable</code> is TRUE, the time counter is incremented by dT.	<code>enable::TRUE</code> or FALSE	none
out	The time counter value, i.e. the time elapsed since the last start and while <code>enable</code> was TRUE is returned.	none	continuous

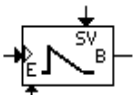
## ascetsdb-etassyse4



### Timer

Timer decrements the time counter by dT and signals when the time counter has reached zero. Its not retriggerable.

Methods	Characteristics	Arguments	Return Value
start	The time counter is set to zero, if the time counter value was previously less than or equal to zero.	<code>startTime::continuous</code>	none
compute	The time counter is decremented by dT.	none	none
out	TRUE is returned, if the time counter is greater then zero. Otherwise FALSE is returned.	none	continuous

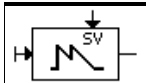


### TimerEnabled

TimerEnabled decrements the time counter by dT and signals when the time counter reaches zero. It must be enabled explicitly.

Methods	Characteristics	Arguments	Return Value
compute	If <code>enable</code> is TRUE, <code>in</code> has a rising edge and the time counter value is less or equal zero, the timer is started, i.e. its counter value is set to the start time. Otherwise the time counter is decremented by dT. If <code>enable</code> is FALSE nothing happens.	<code>enable::TRUE</code> or FALSE <code>in::TRUE</code> or FALSE <code>startTime::continuous</code>	none
out	TRUE is returned, if the time counter value is greater than zero. Otherwise, FALSE is returned.	none	continuous

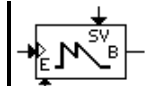
## ascetsdb-etassyse5



### TimerRetrigger

TimerRetrigger decrements the time counter by dT and signal when the time counter has reached zero. It can be retriggered.

Methods	Characteristics	Arguments	Return Value
start	The time counter is set to the start value	startTime::continuous	none
compute	The time counter is decremented by dT.	none	none
out	TRUE is returned, if the time counter value is greater than zero. Otherwise, FALSE is returned.	none	continuous



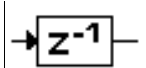
### TimerRetriggerEnabled

TimerRetriggerEnabled decrements the time counter by dT and signals when time counter has reached zero. It can be retriggered and must be enabled explicitly.

Methods	Characteristics	Arguments	Return Value
compute	If enable is TRUE and it has a rising edge, the timer is started i.e. its counter value is set to the start value. Otherwise the time counter is decremented by dT (the time frame). If enable is FALSE nothing happens.	enable::TRUE or FALSE in::TRUE or FALSE startValue::continuous	none
out	TRUE is returned, if the time counter value is greater than zero. Otherwise FALSE is returned.	none	continuous

## ascetsdb-etassyse6

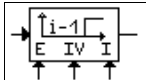
### Delay



#### DelaySignal

DelaySignal delays its input signal by one evaluation step.

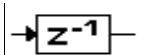
Methods	Characteristics	Arguments	Return Value
compute	The input signal is buffered	signal::TRUE or FALSE	none
out	The buffered signal is returned, thus the input signal is delayed by one step.	none	TRUE or FALSE



#### DelaySignalEnabled

DelaySignalEnabled delays its input signal by one evaluation step. It must be enabled explicitly.

Methods	Characteristics	Arguments	Return Value
reset	If initEnable is TRUE, initValue is buffered	initEnable::TRUE or FALSE initValue::TRUE or FALSE	none
compute	If enable is TRUE, the input signal is buffered.	signal::TRUE or FALSE enable::TRUE or FALSE	none
out	The buffered signal is returned, thus the input signal is delayed by one step.	none	TRUE or FALSE

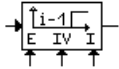


#### DelayValue

DelayValue delays its input value by one evaluation step.

Methods	Characteristics	Arguments	Return Value
compute	The input value is buffered.	value::continuous	none
out	The buffered value is returned, thus the input value is delayed by one step.	none	continuous

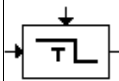
## ascetsdb-etassyse7



### DelayValueEnabled

DelaySignalEnabled delays its input value by one evaluation step. It must be enabled explicitly.

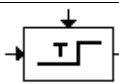
Methods	Characteristics	Arguments	Return Value
reset	If <code>initEnable</code> is TRUE, <code>initValue</code> is buffered.	<code>initEnable</code> ::TRUE or FALSE <code>initValue</code> ::continuous	none
compute	If <code>enable</code> is TRUE, the input value is buffered.	<code>value</code> ::continuous <code>enable</code> ::TRUE or FALSE	none
out	The buffered value is returned, thus the input value is delayed by one step	none	TRUE or FALSE



### TurnOffDelay

TurnOffDelay delays a falling edge of the input signal.

Methods	Characteristics	Arguments	Return Value
compute	A falling edge of the input signal is delayed. If the signal flips from TRUE to FALSE, a timer is started. On being FALSE the timer is incremented by <code>dT</code> and is compared to <code>delayTime</code> . If the input signal is TRUE, the timer is reset.	<code>signal</code> ::TRUE or FALSE <code>delayTime</code> ::continuous	none
out	TRUE is returned if the input signal is TRUE or the timer has not exceeded <code>delayTime</code> . Otherwise FALSE is returned.	none	TRUE or FALSE



### TurnOnDelay

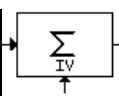
TurnOnDelay delays a rising edge of the input signal.

Methods	Characteristics	Arguments	Return Value
compute	A rising edge of the input signal is delayed. If the signal flips from FALSE to TRUE a timer is started. On being TRUE the timer is incremented by <code>dT</code> and is compared to <code>delayTime</code> . If the input signal is FALSE, the timer is reset.	<code>signal</code> ::TRUE or FALSE <code>delayTime</code> ::continuous	none

### ascetsdb-etassyse8

out	FALSE is returned if the input signal is FALSE, or the timer has not exceeded <code>delayTime</code> . Otherwise, TRUE is returned.	none	TRUE or FALSE
-----	---	------	---------------

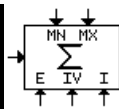
## Memory



### Accumulator

Accumulator adds up its input value.

Methods	Characteristics	Arguments	Return Value
reset	The accumulator is set to <code>initValue</code> .	<code>initValue</code> ::continuous	none
compute	The accumulator is incremented by the input value, i.e. <code>accumulator (new) = accumulator (old) + input value</code> .	<code>value</code> ::continuous	none
out	The accumulator value is returned.	none	continuous



### AccumulatorEnabled

AccumulatorEnabled adds up its input value. It must be enabled explicitly and its accumulator value can be limited.

Methods	Characteristics	Arguments	Return Value
reset	If <code>initEnable</code> is TRUE, the accumulator value is set to <code>initValue</code> .	<code>initValue</code> ::continuous <code>initEnable</code> ::TRUE or FALSE	none
compute	If <code>enable</code> is TRUE, the accumulator is incremented by the input value, i.e. <code>accumulator (new) = accumulator (old) + input value</code> . Additionally, the accumulator value is limited by <code>mn</code> and <code>mx</code> .	<code>value</code> ::continuous <code>mn</code> ::continuous <code>mx</code> ::continuous <code>enable</code> ::TRUE or FALSE	none
out	The accumulator value is returned.	none	continuous

### ascetsdb-etassyse9