SRT / SRAE SOFTWARE DOCUMENTATION SUPER-USER

3. INPUTS

chapter 3

Release : 2.0

REVISIONS DOCUMENT

Release	*Author*	*Date*	*Modifications*
1.0	**M.Mersier**	**04/04/2004**	• **Creation (SRA)**
2.0	**M.Mersier**	**09/08/2006**	• **Update (SRT/SRAE)**

3.1 Analog Inputs

3.1.1 Throttle Position.

3.1.1.1 Throttle Scale.

- Name : **EE.Acq.EchellePap**
- The throttle scale may be expressed in degrees or in %.
- Typic value: 100 for 100 % , or 90 for 90°
- The same parameter is use for Pedal scale

3.1.1.2 Linearisation for throttles position 1 and 2.

The acquired value is linearised using the following formula :

$$\mathbf{FD.ACQ.PAP1 = (AnaPap1Corr + EE.App.Zero\_Pap) * EE.App.GainPap}$$

$$\mathbf{FD.ACQ.PAP2 = (AnaPap2Corr + EE.App.Zero\_Pap2) * EE.App.Gain\_Pap2}$$

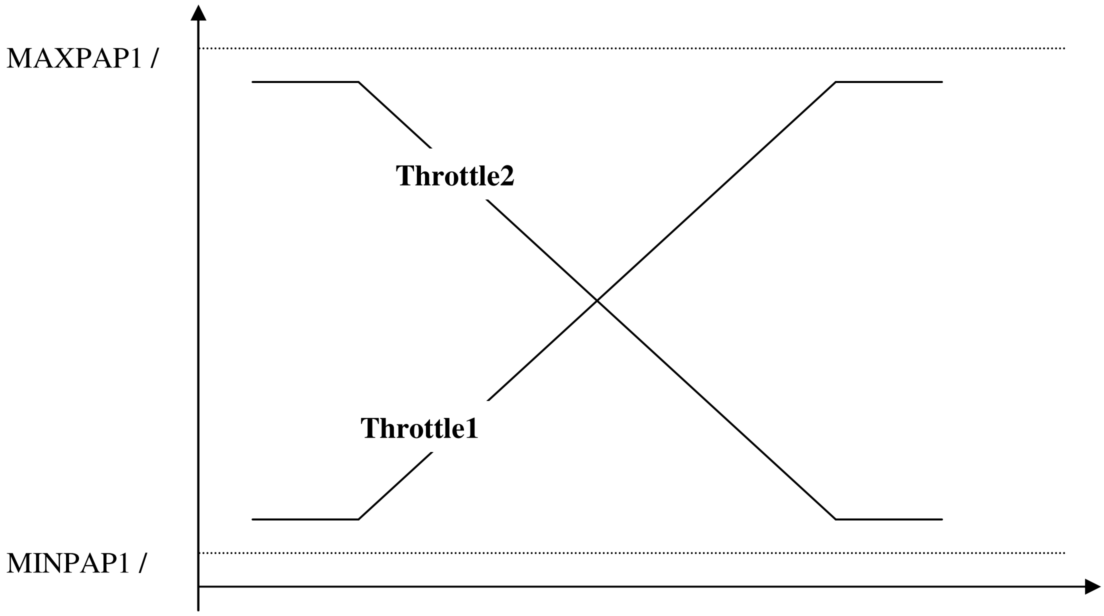
where:

- **FD.ACQ.PAP1 :** Throttle Positions 1 in Degrees (or %)
- **FD.ACQ.PAP2 :** Throttle Positions 2 in Degrees (or %)
- **AnaPap1Corr :** Throttle 1 Acquisition value in Volt
- **AnaPap2Corr :** Throttle 2 Acquisition value in Volt
- **EE.App.Zero_Pap :** Value in Volts of the learning position 0° (or 0%) of throttle 1 sensor..
- **EE.App.Zero_Pap2 :** Value in Volts of the learning position 0° (or 0%) of throttle 2 sensor..
- **EE.App.Gain_Pap :** Learning Gain in °/Volt to obtain 90° (or 100%) when throttle 1 is fully open.
- **EE.App.Gain_Pap2 :** Learning Gain in °/Volt to obtain 90° (or 100%) when throttle 2 is fully open.

3.1.1.3 Diagnostic.

3.1.1.3.1 Using in motorised throttle mode.

When using a motorised throttle , the both throttle signal are :



3.1.1.3.2 Validity Criteria.

The acquired values of the throttle position inputs, if connected (Throttle Input <> 0), are first tested against maximum **EE.Acq.SAnaPap_Max** and minimum **EE.Acq.SAnaPap_Min** values to verify their validity.

3.1.1.3.3 Coherence Criteria (only in motorised throttle mode).

If the both throttle are valid and if the coherence test is valid by the map **EE.Acq.AutoCoherencePAP** , a fault throttle coherence is declare when **Throttle1+Throttle2** < **EE.Acq.EchellePap** - **EE.Acq.EcPapCoherence**,.

3.1.1.3.4 Default Detection.

Should the measured value be outside these limits for a specified number of consecutive acquisitions **EE.Acq.NbConfChPap** , a fault is signalled in the relevant diagnostic indicator

3.1.1.3.5 Default Mode.

- In case of two potentiometers being fitted, the throttle position **FD.Acq.Pap** is initially taken from channel 1.
- In case of default on channel 1, the throttle position **FD.Acq.Pap** is taken from channel 2 (if fitted), and the Coherence Criteria is not tested.
- If channel 2 is in default and channel 1 is not, then throttle position **FD.Acq.Pap** is taken back from channel 1 and the Coherence Criteria is not tested.
- In case of coherence default , then throttle position **FD.Acq.Pap** is taken from the minus channel
- In case where both channels are in default, and the RCO of the throttle motorised is write to 0 (there is no Hbridge command and the throttle comeback to its Limphome position for Safety function) :

FD.Acq.Pap = EE.Acq.PapParDefaut

3.1.1.4 Breakpoints.

3.1.1.4.1 Number of breakpoints

The table **EE.CfgU.NbkpThrottle** defines the number of breakpoints for tables function of the throttle position. The maximal value is 24.

3.1.1.4.2 Breakpoint Table

The table **EE.Bkps.Throttle** define the breakpoints (quantity defined above) expressed in degrees (or %) for addressing maps as a function of throttle position.

3.1.1.4.3 Throttle Zone Breakpoints.

The table **EE.Bkps.ThrottleShort** contains 5 breakpoints expressed in degrees and is used for addressing maps as a function of throttle zones (these 5 breakpoints define 7 zones, including below b.p. 1 and over b.p. 5).

3.1.2 Pedal Position (in motorised throttle mode ONLY).

3.1.2.1 Linearisation for Pedals position 1 and 2 (Use only for motorised throttle).

The acquired value is linearised using the following formula :

$$\mathbf{FD.Acq.Pdl_i = (AnaPdl_iCorr - EE.App.Zero\_Pdl_i) * EE.App.Gain\_Pdl_i}$$

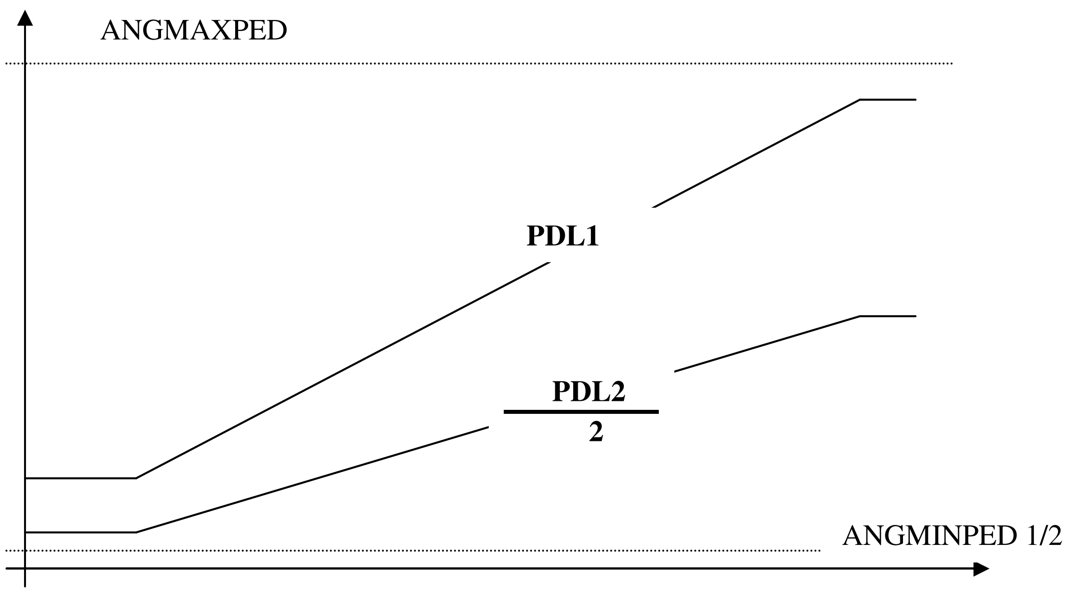
where:

- **FD.ACQ.PDL*i* :** Pedal Positions 1 and 2 in Degrees (or %)
- **AnaPdl*i*Corr :** Pedal 1 and 2 Acquisition value in Volt
- **EE.App.Zero_Pdl*i* :** Value in Volts of the learning position 0° (or 0%) of pedal 1 and 2 sensors.. .
- **EE.App.Gain_Pdl*i* :** Learning Gain in °/Volt to obtain 90° (or 100%) when pedal 1 and 2 is fully open.

with ***i*** : 1 or 2 depending on valid channel.

3.1.2.2 Diagnostic.

When using an electrical pedal for a motorised throttle , the both pedal signal are :



3.1.2.2.1 Validity Criteria.

The acquired values of the pedal position inputs, if connected (**EE.Acq.Pdl1Input** et **EE.Acq.Pdl2Input** <> 0), are first tested against maximum (**EE.Acq.SAnaPdl_Max**) and minimum (**EE.Acq.SAnaPdl_Min**) to verify their validity.

3.1.2.2.2 Coherence Criteria .

If the both pedals are valid , and if the map ***EE.Pmot.Version_Banc*** is disable ,a fault pedal coherence is declare when **Pdl1-Pdl2 < EE.Acq.EcartPdl** ,.

NOTE: The map ***EE.Pmot.Version_Banc*** allow to connect a standard potentiometer on Pdl 1 input instead an electrical pedal on Pdl1 and Pdl2 , only for engine bench using .

3.1.2.2.3 Default Detection.

Should the measured value be outside these limits for a specified number of consecutive acquisitions (**EE.Acq.NbConfChPdl**), a fault is signaled in the relevant diagnostic indicator

3.1.2.2.4 Default Mode.

- The pedal position is initially taken from channel 1.
- In the case of default on channel 1, the pedal position is taken from channel 2 .
- If channel 2 is in default and channel 1 is not, then pedal position is taken back from channel 1.
- In the case of coherence default , then pedal position is taken from the minus channel
- In the case where both channels are in default:

FD.Acq.Pdl = PdlParDefaut

3.1.2.3 Breakpoints.

3.1.2.3.1 Number of breakpoints

The table **EE.CfgU.NbkpPdl** defines the number of breakpoints for tables function of the pedal position. The maximal value is 24.

3.1.2.3.2 Breakpoint Table

The table **EE.Bkps.Pedale** define the breakpoints (quantity defined above) expressed in degrees is used for addressing maps as a function of pedal position.

3.1.3 Learning Throttle and Pedal Position.

3.1.3.1 Mechanical Throttle

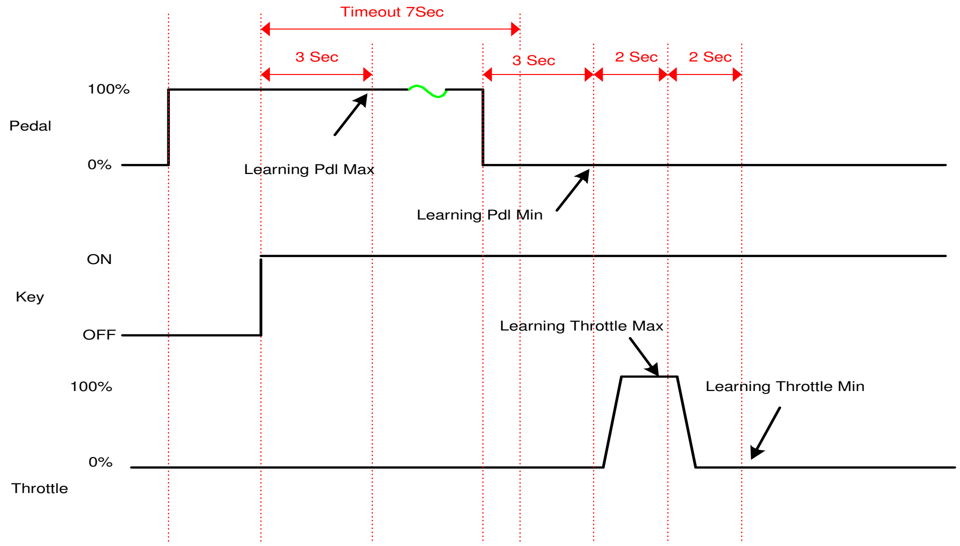
The acquisition of gains (**EE.App.Gain_Pap , EE.App.Gain_Pap2)** and zeros (**EE.App.Zero_Pap , EE.App.Zero_Pap2)** for the throttle inputs may be done automatically by successively positioning the throttles in the two positions defined below (typically 0°/0% and 90°/100%), and using VISION to write the values of 1 (in one extreme position) and then 2 (in the other extreme position) in the variable "**LearnThr**" .

NOTE 1: to be sure to always have a repetitive situation (at the dyno & on various engines on the track), it is recommended to remove all mechanical pre-opening prior to doing this learning, and then put it back afterwards.

NOTE 2: the learning must always be done in one go, first the zero, then the full throttle. If, for some reason, the throttle wasn't properly set when one of the positions was learnt, don't try to redo it partially (redoing several times the zero or several times the full throttle), finish the procedure, even with a wrong position, and redo it properly afterwards: zero first, then full throttle.

3.1.3.2 Motorised Throttle

The acquisition of gains (EE.App.Gain_Pap , EE.App.Gain_Pap2) and zeros (EE.App.Zero_Pap , EE.App.Zero_Pap2) for the throttle and gains (EE.App.Gain_Pdl1 , EE.App.Gain_Pdl2) and zeros (EE.App.Zero_Pdl1 , EE.App.Zero_Pdl2) pedal inputs may be done automatically by pushing the pedal thoroughly before Key On , and release the pedal (3 Sec Min and 7 Sec Max) after Key On



3.1.4 Inlet (manifold) Air Pressure.

3.1.4.1 Linearisation.

During normal operation the acquired value is linearised using the following formula:

$$\text{PadmBase} = (((\text{FD.Vrs.Padm1} + \text{FD.Vrs.Padm2} + \text{FD.Vrs.Padm3} + \text{FD.Vrs.Padm4})/4) - \text{EE.Acq.zero\_padm}) * \text{EE.Acq.gain\_padm}$$

where:

- ***PadmBase :*** Inlet Air pressure acquire value in mbar.
- ***FD.Vrs.Padm1..4*** : Inlet Air pressure acquisition in Volt on each TDCs.
- ***EE.Acq.zero_padm*** : Zero sensor in Volt.
- ***EE.Acq.gain_padm*** : Gain sensor in mbar / Volt.

3.1.4.2 Diagnostic.

3.1.4.2.1 Validity Criteria.

The acquired value of the inlet air pressure input, if connected (**EE.CfgU.InletPressInput** <> 0,), is first tested against maximum **EE.Acq.SAnaPadm_Max** and minimum **EE.Acq.SAnaPadm_Min** to verify its validity.

3.1.4.2.2 Default Mode

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator and:

- In the case of RPM / Throttle function or throttle signal not valid:

$$\text{FD.Acq.Padm} = \text{EE.Acq.PadmParDefaut}$$

- In the case of RPM / Pressure function and the throttle signal is valid, the inlet air pressure is reconstituted from a two dimensional map as a function of RPM and throttle position. (This table is described in the section Injection Time Calculation from documentation Injection.doc) :

$$\text{FD.Acq.Padm} = \text{Interpolation2D(EE.Acq.DefPadm}, f(\text{Papillon}, \text{Régime}))$$

Where:

- ***EE.Acq.DefPadm*** in mBar (max : 4080mbars).

3.1.4.3 Filtering.

The acquired value is filtered using the following formula:

$$\boxed{FD.Acq.Padm = FD.Acq.Padm + ((PadmBase - FD.Acq.Padm) * EE.Acq.KfiltPatm)}$$

Where:

- ***PadmBase :*** Inlet Air pressure linearised value in mbar.
- ***FD.Acq.Padm*** : Inlet Air pressure in mBar.
- ***EE.Acq.KfiltPadm*** : Coefficient filter (between 0 and 1).

3.1.4.4 Breakpoints.

3.1.4.4.1 Number of breakpoints for the Inlet Air Pressure.

The table **EE.CfgU.NbkpInletPress** defines the number of breakpoints for tables function of the Inlet Air Pressure. The maximal value is 24.

3.1.4.4.2 Breakpoint Table

The table **EE.Bkps.InletP** define the breakpoints (quantity defined above) expressed in mbar is used for addressing maps as a function of inlet air pressure.

3.1.5 Barometric Pressure.

3.1.5.1 Sensor Mode.

3.1.5.1.1 Linearisation.

If barometric pressure sensor is fitted the acquired value is linearised permanently (or engine stopped only), using the following formula:

$$\boxed{PatmBase = (AnaPatmCorr - EE.Acq.zero\_patm) * EE.Acq.gain\_patm}$$

where:

- ***PatmBase*** : Barometric pressure linearised value in mbar.
- ***AnaPatmCorr :*** Barometric Pressure acquisition in Volt.
- ***EE.Acq.zero_patm*** : Zero sensor in Volt .
- ***EE.Acq.gain_patm*** : Gain sensor in mbar / Volt.

3.1.5.1.2 Diagnostic.

3.1.5.1.2.1 Validity Criteria

The acquired value of the barometric pressure input, if connected (**EE.CfgU.BaroPressInput** <> 0,), is first tested against maximum **EE.Acq.SAnaPatm_Max** and minimum **EE.Acq.SAnaPatm_Min** to verify its validity:

3.1.5.1.2.2 Default Mode

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator and:

$$\boxed{\textbf{FD.Acq.Patm} = \textbf{PatmParDefaut}}$$

3.1.5.1.3 Filtering.

The acquired value is filtered permanently (or engine stopped only), using the following formula:

$$\boxed{\textbf{FD.Acq.Patm} = \textbf{FD.Acq.Patm} + (\textbf{PatmBase} - \textbf{FD.Acq.Patm}) * \textbf{EE.Acq.KfiltPatm}}$$

Where:
- ***PatmBase :*** Barometric pressure linearised value in mbar.
- ***FD.Acq.Patm*** : Barometric pressure in mBar.
- ***EE.Acq.KfiltPatm*** : Coefficient filter (between 0 and 1).

3.1.5.2 No Sensor Mode.

In the case where barometric pressure sensor is not fitted (Barometric Pressure Input = 0,), the barometric pressure value is calculated from the inlet air pressure (if one is fitted) as follows:

$$\boxed{\textbf{FD.Acq.Patm} = \textbf{FD.Acq.Patm} + (\textbf{FD.Acq.Padm} - \textbf{FD.Acq.Patm}) * \textbf{EE.Acq.KfiltPatm}}$$

Only in the following conditions:

(RPM = 0) or (RPM < EE.Acq.Sreg_Patm and Throttle >= EE.Acq.Spap_Patm) for longer than EE.Acq.Stab_Patm mSec ..

Where:
- ***RPM, EE.Acq.Sreg_Patm*** in rpm.
- ***Throttle, EE.Acq Spap_Patm*** in Degrees.
- ***EE.Acq Stab_Patm*** in milliseconds

3.1.5.3 Barometric Pressure Initialisation, Engine ON.

In the case where updating of the acquisition is not authorised when the engine is ON, barometric pressure is initialised with the “current” value backed-up from the last time the ECU was powered up. This is so that it can take a correct “current” value in the case of an ECU power up while the engine is already turning (power failure inducing an ECU reset while the vehicle is running).
The selection barometric pressure acquisition when engine ON or Stopped is **EE.Acq.PatmAlways.**

3.1.6 Fuel Pressure.

3.1.6.1 Linearisation.

During normal operation the acquired value is linearised using the following formula:

$$\boxed{PessBase = (AnaPessCorr - EE.Acq.zero\_pess) * EE.Acq.gain\_pess}$$

Where:

- ***PessBase*** : Fuel pressure linearised value in mbar.
- ***AnaFuelCorr :*** Fuel Pressure acquisition in Volt.
- ***EE.Acq.zero_pess*** : Zero sensor in Volt
- ***EE.Acq.gain_pess*** : Gain sensor in mBar / Volt.

3.1.6.2 Diagnostic.

3.1.6.2.1 Validity Criteria

The acquired value of the fuel pressure input, if connected (**EE.CfgU.FuelPressInput**<> 0,), is first tested against a minimum value **EE.Acq.SAnaPess_Min** when the RPM is greater than the threshold **EE.Acq.SAnaPess_Rpm** and a maximum value **EE.Acq.SAnaPess_Max** to verify its validity.

3.1.6.2.2 Default Mode or Channel not use.

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator and :

$$\boxed{FD.Acq.Pess = EE.Acq.PfuelParDefaut}$$

3.1.6.3 Filtering.

The acquired value is filtered using the following formula: :

$$\boxed{FD.Acq.Pess = FD.Acq.Pess + (PessBase - FD.Acq.Pess) * EE.Acq.KfiltPess}$$

Where:

- ***PessBase :*** Fuel pressure linearised value in mbar.
- ***FD.Acq.Pess*** : Fuel pressure in mBar.
- ***EE.Acq.KfiltPess*** : Coefficient filter (between 0 and 1).

3.1.6.4 Breakpoints.

3.1.6.4.1 Number of breakpoints of fuel pressure.

The table **EE.CfgU.NbkpFuelPress** defines the number of breakpoints for tables function of the fuel pressure. The maximal value is 24.

3.1.6.4.2 Breakpoint Table.

The table **EE.Bkps.FuelP** define the breakpoints (quantity defined above) expressed in Bar is used for addressing maps as a function of fuel pressure.

3.1.7 Oil Pressure.

3.1.7.1 Linearisation.

During normal operation the acquired value is linearised using the following formula:

$$\mathbf{PoilBase = (AnaPoilCorr - EE.Acq.zero\_poil) * EE.Acq.gain\_poil}$$

Where:

- ***PoilBase*** : Oil pressure linearised value in mbar.
- ***AnaPoilCorr :*** Oil Pressure acquisition in Volt.
- ***EE.Acq.zero_poil*** : Zero sensor in Volt.
- ***EE.Acq.gain_poil*** : Gain sensor in mbar / Volt.

3.1.7.2 Diagnostic.

3.1.7.2.1 Validity Criteria

The acquired value of the oil pressure input, if connected **EE.CfgU.OilPressInput<>** 0,), is first tested against a minimum value **EE.Acq.SAnaPoil_Min** when the RPM is greater than the threshold **EE.Acq.SAnaPoil_Rpm** and a maximum value **EE.Acq.SAnaPoil_Max** to verify its validity:.

3.1.7.2.2 Default Mode or Channel not use.

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator
:

$$\boxed{FD.Acq.Poil = EE.Acq.PoilParDefaut}$$

3.1.7.3 Filtering.

The acquired value is filtered using the following formula:

$$\boxed{FD.Acq.Poil = FD.Acq.Poil + (PoilBase - FD.Acq.Poil) * EE.Acq.KfiltPoil}$$

Where:

- ***PoilBase :*** Oil pressure linearised value in mbar**.**
- ***FD.Acq.Poil*** : Oil pressure in mBar.
- ***EE.Acq.KfiltPoil*** : Coefficient filter (between 0 and 1).

3.1.8 Water Temperature.

3.1.8.1 Diagnostic.

3.1.8.1.1 Validity Criteria.

The acquired value of the water temperature input, if connected (**EE.CfgU.WaterTempInput<>** 0,), is first tested against maximum **EE.Acq.SAnaTeau_Max** and minimum **EE.Acq.SAnaTeau_Min** to verify its validity:

3.1.8.1.2 Default Mode or Channel not use.

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator and :

$$\boxed{FD.Acq.Teau = EE.Acq.TeauParDefaut}$$

3.1.8.2 Linearisation.

3.1.8.2.1 CTN Sensor.

When CTN input is selected, an interpolation in the table **EE.Acq.CtnWaterT.Bkpts :**for) is used for Ohms to °C conversion.

This table contain 24 values of the CTN resistor used from –40°C to +190°C by step of 10°C.

3.1.8.2.2 PT1000 Sensor.

When PT1000 input is selected, a mathematical model is used for Ohms to °C conversion .

3.1.8.3 Filtering.

The acquired value filtered using the following formula:

$$\boxed{\mathbf{FD.Acq.Teau = FD.Acq.Teau + (TeauBase - FD.Acq.Teau) * EE.Acq.KfiltTeau}}$$

Where:

- ***TeauBase :*** Water Temperature linearised value in °C.
- ***FD.Acq.Teau***: Water Temperature in °C.
- ***EE.Acq.KfiltTeau*** : Coefficient filter (between 0 and 1).

3.1.8.4 Breakpoints.

3.1.8.4.1 Number of breakpoints for the water temperature.

The table **EE.CfgU.NbkpWaterTemp** defines the number of breakpoints for tables function of the water temperature. The maximal value is 24.

3.1.8.4.2 Breakpoint Table.

The table **EE.Bkps.WaterT** define the breakpoints (quantity defined above) expressed in °C is used for addressing maps as a function of water temperature.

3.1.9 Air Temperature.

3.1.9.1 Diagnostic

3.1.9.1.1 Validity Criteria.

The acquired value of the air temperature input, if connected (**EE.CfgU.AirTempInput<>** 0,), is first tested against maximum **EE.Acq.SAnaTair_Max** and minimum **EE.Acq.SAnaTair_Min** to verify its validity.

3.1.9.1.2 Default Mode or Channel not use.

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator and :

$$\boxed{\mathbf{FD.Acq.Tair = EE.Acq.TairParDefaut}}$$

3.1.9.2 Linearisation.

3.1.9.2.1 CTN Sensor.

When CTN input is selected, an interpolation in the table **EE.Acq.CtnAirT.Bkpts :**Airfor Air)is used for Ohms to °C conversion .

This table contain 24 values of the CTN resistor used from –40°C to +190°C by step of 10°C.

3.1.9.2.2 PT1000 Sensor.

When PT1000 input is selected, a mathematical model is used for Ohms to °C conversion .

3.1.9.3 Filtering.

The acquired value filtered using the following formula:

$$\boxed{\mathbf{FD.Acq.Tair = FD.Acq.Tair + (TairBase - FD.Acq.Tair) * EE.Acq.KfiltTair}}$$

Where:

- ***TairBase :*** Air Temperature linearised value in °C.
- ***FD.Acq.Tair***: Air Temperature in °C.
- ***EE.Acq.KfiltTair*** : Coefficient filter (between 0 and 1).

3.1.9.4 Breakpoints.

3.1.9.4.1 Number of breakpoints for the air temperature.

The table **EE.CfgU.NbkpAirTemp** defines the number of breakpoints for tables function of the air temperature. The maximal value is 24.

3.1.9.4.2 Breakpoint Table.

The table **EE.Bkps.AirT** define the breakpoints (quantity defined above), expressed in °C, is used for addressing maps as a function of air temperature.

3.1.10 Oil Temperature.

3.1.10.1 Diagnostic.

3.1.10.1.1 Validity Criteria.

The acquired value of the oil temperature input, if connected (**EE.CfgU.OilTempInput<>** 0,), is first tested against maximum (**EE.Acq.SAnaToil_Max**) and minimum (**EE.Acq.SAnaToil_Min** to verify its validity.

3.1.10.1.2 Default Mode or Channel not use.

Should the value be outside these limits, a fault is signaled in the relevant diagnostic indicator and :

$$\boxed{\mathbf{FD.Acq.Toil = EE.Acq.ToilParDefaut}}$$

3.1.10.2 Linearisation.

3.1.10.2.1 CTN Sensor.

When CTN input is selected, an interpolation in the table **EE.Acq.CtnOilT.Bkpts** :Oilfor Oil)is used for Ohms to °C conversion .

This table contain 24 values of the CTN resistor used from –40°C to +190°C by step of 10°C.

3.1.10.2.2 PT1000 Sensor.

When PT1000 input is selected, a mathematical model is used for Ohms to °C conversion .

3.1.10.3 Filtering.

The acquired value filtered using the following formula:

$$\boxed{\mathbf{FD.Acq.Toil = FD.Acq.Toil + (ToilBase - FD.Acq.Toil) * EE.Acq.KfiltToil}}$$

Where:

- ***ToilBase :*** Oil Temperature linearised value in °C.
- ***FD.Acq.Toil***: Oil Temperature in °C.
- ***EE.Acq.KfiltToil*** : Coefficient filter (between 0 and 1).

3.1.11 Fuel Temperature.

3.1.11.1 Diagnostic.

3.1.11.1.1 Validity Criteria.

The acquired value of the fuel temperature input, if connected (**EE.CfgU.FuelTempInput<>** 0,), is first tested against maximum **EE.Acq.SAnaTfuel_Max** and minimum **EE.Acq.SAnaTfuel_Min** to verify its validity.

3.1.11.1.2 Default Mode or Channel not use.

Should the value be outside these limits, a fault is signaled in the relevant diagnostic indicator and :

$$\boxed{\mathbf{FD.Acq.Tess = EE.Acq.TessParDefaut}}$$

3.1.11.2 Linearisation.

3.1.11.2.1 CTN Sensor.

When CTN input is selected, an interpolation in the table **EE.Acq.CtnFuelT.Bkpts :**Oilfor Oil)is used for Ohms to °C conversion .

This table contain 24 values of the CTN resistor used from –40°C to +190°C by step of 10°C.

3.1.11.2.2 PT1000 Sensor.

When PT1000 input is selected, a mathematical model is used for Ohms to °C conversion .

3.1.11.3 Filtering.

The acquired value filtered using the following formula:

$$\boxed{\mathbf{FD.Acq.Tess = FD.Acq.Tess + (TessBase - FD.Acq.Tess) * EE.Acq.KfiltTess}}$$

Where:

- ***TessBase :*** Fuel Temperature linearised value in °C.
- ***FD.Acq.Tess***: Fuel Temperature in °C.
- ***EE.Acq.KfiltTess*** : Coefficient filter (between 0 and 1).

3.1.11.4 Breakpoints.

3.1.11.4.1 Number of breakpoints for the fuel temperature.

The table **EE.CfgU.NbkpFuelTemp** defines the number of breakpoints for tables function of the fuel temperature. The maximal value is 24.

3.1.11.4.2 Breakpoint Table.

The table **EE.Bkps.FuelT** define the breakpoints (quantity defined above), expressed in °C, is used for addressing maps as a function of fuel temperature.

3.1.12 Thermocouple1.

3.1.12.1 Thermocouple impedance N°1.

- Name : **SYSeep.Tck1Parms.SensorImpedance**
- This parameter (unit: kOhms) is in intern used for given the correct value in mVolts from thermocouple input .

3.1.12.2 Linearisation .

During normal operation the acquired value is linearised using the following formula:

$$\begin{aligned}\mathbf{TCK1Base} = &(\mathbf{AnaTck1.Corr} * \mathbf{SYSeep.Tck1Parms.Gain}) \\ &- \mathbf{SYSeep.Tck1Parms.Zero} + \mathbf{Inputs.TBox.Corr}\end{aligned}$$

The thermocouple input is not compensated, but adjusted to the internal temperature of the ECU (Tbox).

Where:

- ***Tck1Base***: Tck1 Temperature linearised value in °C.
- ***AnaTck1.Corr :*** Tck1 Temperature acquisition in Volt.
- ***SYSeep.Tck1Parms.Zero*** : Zero sensor in Volt.
- ***SYSeep.Tck1Parms.Gain***: Gain sensor in °C / Volt.
- ***Input.TBox.Corr*** : Internal Box Temperature °C.

3.1.12.3 Diagnostic.

3.1.12.3.1 Validity Criteria.

The acquired raw value of the input thermocouple 1, is first tested against maximum **EE.Acq.SAnaTck_Max** and minimum **EE.Acq.SAnaTck_Min** to verify its validity (note that the lower threshold is a bit difficult to adjust since at ambient temperature, the signal is very close to 0V.

3.1.12.4 Filtering.

the acquired corrected value is filtered using the following formula:

$$\mathbf{FD.Acq.Tck1} = \mathbf{FD.Acq.Tck1} + (\mathbf{Tck1base} - \mathbf{FD.Acq.Tck1}) * \mathbf{EE.Acq.KfiltTck}$$

Where:

- ***Tck1Base :*** Tck1 Temperature linearised value in °C.
- ***FD.Acq.Tck1***: Tck1 Temperature in °C.
- ***EE.Acq.KfiltTck*** : Coefficient filter (between 0 and 1).

3.1.13 Thermocouple2.

3.1.13.1 Thermocouple impedance N°2.

- Name : **SYSeep.Tck2Parms.SensorImpedance**
- This parameter (unit: kOhms) is in intern used for given the correct value in mVolts from thermocouple input .

3.1.13.2 Linearisation.

During normal operation the acquired value is linearised using the following formula:

$$\begin{aligned}\mathbf{TCK2Base} = &(\mathbf{AnaTck2.Corr} * \mathbf{SYSeep.Tck2Parms.Gain}) \\ &- \mathbf{SYSeep.Tck2Parms.Zero} + \mathbf{Inputs.TBox.Corr}\end{aligned}$$

The thermocouple input is not compensated, but adjusted to the internal temperature of the ECU (Tbox).

Where:

- ***Tck2Base***: Tck2 Temperature linearised value in °C.
- ***AnaTck2.Corr :*** Tck2 Temperature acquisition in Volt.
- ***SYSeep.Tck2Parms.Zero*** : Zero sensor in Volt.
- ***SYSeep.Tck2Parms.Gain***: Gain sensor in °C / Volt.
- ***Input.TBox.Corr*** : Internal Box Temperature °C.

3.1.13.3 Diagnostic.

3.1.13.3.1 Validity Criteria.

The acquired raw value of the input thermocouple 1, is first tested against maximum **EE.Acq.SAnaTck_Max** and minimum **EE.Acq.SAnaTck_Min** to verify its validity (note that the lower threshold is a bit difficult to adjust since at ambient temperature, the signal is very close to 0V.

3.1.13.4 Filtering.

The acquired corrected value is filtered using the following formula:

$$\boxed{FD.Acq.Tck2 = FD.Acq.Tck2 + (Tck2base - FD.Acq.Tck2) * EE.Acq.KfiltTck2}$$

Where:

- ***Tck2Base :*** Tck2 Temperature linearised value in °C.
- ***FD.Acq.Tck2***: Tck2 Temperature in °C.
- ***EE.Acq.KfiltTck2*** : Coefficient filter (between 0 and 1).

3.1.14 Gearbox Position.

3.1.14.1 Diagnostic.

3.1.14.1.1 Validity Criteria.

The acquired value of the gear position input, if connected (**EE.CfgU.BarrelInput<>** 0,), is first tested against maximum **EE.Acq.SAnaBarrel_Max**Box and minimum **EE.Acq.SAnaBarrel_Min** to verify its validity.

3.1.14.1.2 Default Mode or Channel not use.

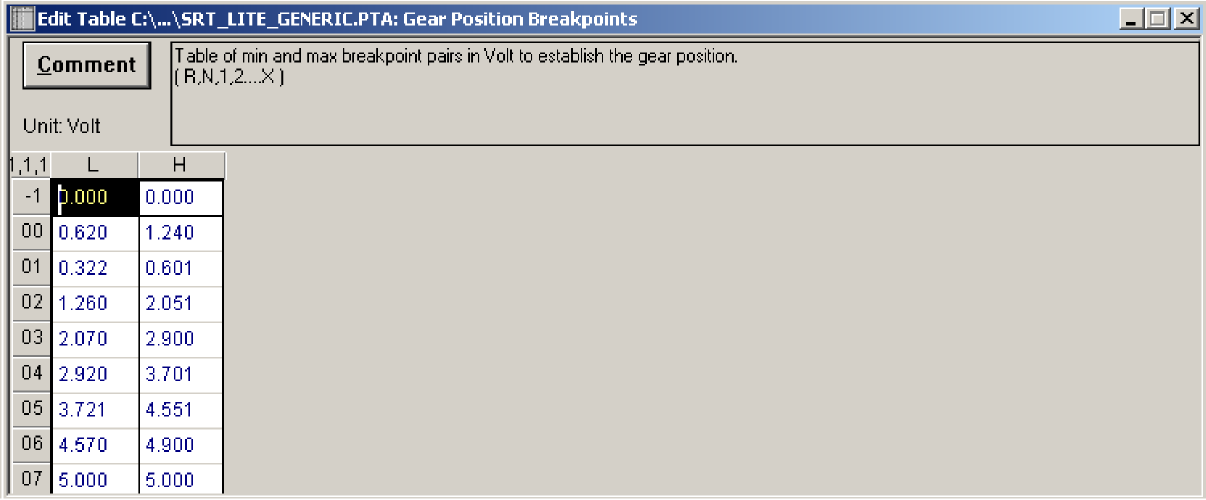
Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator. If a fault exists or a sensor is not fitted::

$$\boxed{RapBv = EE.Acq.Rapport\_defaut}$$

In case of rotactor sensor (**EE.CfgU.TypeBarrel** :: = 1), the default mode is applied only when the default is present for over a defined period of time (same time as the “timeout on gear change” described in the paragraph below), in order to avoid problems happening during the “open circuit” situation (shift from one contact of the rotactor to the next).

3.1.14.2 Gear elaboration, normal mode.

The table **EE.Acq.Seuil_RapBv** contains 9 pairs of values in Volt (minimum and maximum) defining the stable electric positions of the gear sensor for the various gear positions. The breakpoints of the table are defined in the order R, N, 1…6 or 7, but the order of the electrical pairs is free (which allows for “exotic” sensors or gearbox configuration).
The values of each pair must be in increasing voltage order.
The values of each pair must not overlap the values of another pair.
Unused gear positions must be filled-in with 0.000V (or 5.000V).
The free intervals between the pairs correspond to positions where the gear is indeterminate.
In the example below, the calibrated gearbox is: 1-N-2-3-4-5-6 (no reverse and no 7th gear).



Edit Table C:\...\SRT_LITE_GENERIC.PTA: Gear Position Breakpoints

Comment

Table of min and max breakpoint pairs in Volt to establish the gear position.
(R,N,1,2....X)

Unit: Volt

1,1,1	L	H
-1	0.000	0.000
00	0.620	1.240
01	0.322	0.601
02	1.260	2.051
03	2.070	2.900
04	2.920	3.701
05	3.721	4.551
06	4.570	4.900
07	5.000	5.000

If the gear sensor remains in an indeterminate position for longer than the timeout **EE.Acq.T_out_rapport** then:

$$\boxed{\textbf{RapBv} = \textbf{EE.Acq.Rapport\_defaut}}$$

The gear position **FD.Acq.Rapport_ascii** is expressed in ASCII value for dashboard and vision display .

- In case of fixed gear , the ASCII value is read in the map **EE.Acq.asciiraprap**

- In case of default , the ASCII value is read in the map **EE.Acq.asciirap** if the map **EE.Acq.AsciiRapTypeDef** is > 0 , either the ASCII value is read for electrical default in the map **EE.Acq.AsciiRapDef[0]** and for timeout default in the map **EE.Acq.AsciiRapDef[1.**

3.1.15 Battery Voltage.

3.1.15.1 Diagnostic.

3.1.15.1.1 Validity Criteria.

The acquired value of the input battery voltage is first tested against a minimum value **EE.Acq.SAnaVbatt_Min ,** when the RPM is greater than the threshold **EE.Acq.SAnaVbatt_rpm** , to verify its validity.

Should the value be below this limit while the RPM is over the specified threshold, a fault is signaled in the relevant diagnostic indicator.

3.1.15.2 Filtering.

The acquired value is filtered using the following formula:

$$FD.Acq.Vbatt = FD.Acq.Vbatt + (Vbatt.Corr - FD.Acq.Vbatt) * EE.Acq.KfiltVbatt$$

Where:

- **Vbatt.Corr :** Battery voltage acquisition in Volt
- **FD.Acq.Vbatt** Battery voltage filtered in Volt.
- **EE.Acq.KfiltVbatt** Coefficient filter (between 0 and 1).

3.1.16 Lambda1.

3.1.16.1 Filtering.

The acquired value is filtered using the following formula:

$$FD.Acq.Lambda1 = FD.Acq.Lambda1 + (Lambda1Corr - FD.Acq.Lambda1) * EE.Acq.KfiltLbd$$

Where:

- ***Lambda1.Corr*** *:* Lambda1 acquisition value in Volt.
- ***FD.Acq.Lambda1*** : Lambda1 value in Volt.
- ***EE.Acq.KfiltLbd*** : Coefficient filter (between 0 and 1).

3.1.16.2 Air/Fuel 1 Ratio Linearisation

The Air/Fuel Ratio **FD.Acq.AirFuel1** is linearised by interpolation in the table **EE.Acq.air_fuel**, using a 33 equi-spaced breakpoint F(FD.Acq.Lambda1) (0.0 to 5.0 Volt).

3.1.17 Lambda2.

3.1.17.1 Filtering.

The acquired value is filtered using the following formula:

$$FD.Acq.Lambda2 = FD.Acq.Lambda2 + (Lambda2Corr - FD.Acq.Lambda2) * EE.Acq.KfiltLbd$$

Where:

- ***Lambda2.Corr*** *:* Lambda2 acquisition value in Volt.
- ***FD.Acq.Lambda2*** : Lambda2 value in Volt.
- ***EE.Acq.KfiltLbd*** : Coefficient filter (between 0 and 1).

3.1.17.2 Air/Fuel 2 Ratio Linearisation

The Air/Fuel Ratio **FD.Acq.AirFuel2** is linearised by interpolation in the table **EE.Acq.air_fuel**, using a 33 equi-spaced breakpoint F(FD.Acq.Lambda2) (0.0 to 5.0 Volt).

3.1.18 Cockpit Advance Trimmer

3.1.18.1 Index Cockpit Advance Trimmer Elaboration

The table **EE.Ign.BornesClic** contains 5 pairs of values in Volt (minimum and maximum) defining the electrical limits of each trimmer position for the 5 index positions. **FD.Ign.IxClicAva** is the number position recognised.
The values of each pair must be in increasing voltage order.
The values of each pair must not overlap the values of another pair.
Unused positions (if any) should be filled-in with electrical values impossible to reach (6V for instance)
.
The free intervals between the pairs correspond to error positions , and the index position **FD.Ign.IxClicAva** is clear to 0.

3.1.19 Cockpit Injection Trimmer

3.1.19.1 Index Cockpit Injection Trimmer Elaboration

The table **EE.Ti.BornesClicRic Ric** contains 5 pairs of values in Volt (minimum and maximum) defining the electrical limits of each trimmer position for the 5 index positions. **FD.Acq.IxTabClicRic** is the number position recognised.
The values of each pair must be in increasing voltage order.
The values of each pair must not overlap the values of another pair.
Unused positions (if any) should be filled-in with electrical values impossible to reach (6V for instance)
.
The free intervals between the pairs correspond to error positions , and the index position **FD.Acq.IxTabClicRic** keep its previous value.

3.1.20 Start Limiters Trimmer

3.1.20.1 Index Start Limiters Trimmer Elaboration

The table **EE.Acq.RotLimDep_Thd** contains 5 pairs of values in Volt (minimum and maximum) defining the electrical limits of each trimmer position for the 5 index positions.
FD.Acq.RotacLimDep is the number position recognised.
The values of each pair must be in increasing voltage order.
The values of each pair must not overlap the values of another pair.

Unused positions (if any) should be filled-in with electrical values impossible to reach (6V for instance)
.
The free intervals between the pairs correspond to error positions . During **EE.Acq.RotacLimDep_T_HS _T_HS** (mSmSec the index position **FD.Acq.RotacLimDep** keep its previous value . If trimmer is on error position since more than this timeout, the index position **FD.Acq.RotacLimDep** is fixed to the default value **EE.Acq.RotacLimDep_Default _Default** and a fault is signalled in the relevant diagnostic indicator.

3.2 Switch Inputs

3.2.1 SW Pit Limiter.

The input is declared active to ground or active to Vref by the map **EE.SW.PitLim.LevelON**
- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.PitLim.ConfTime.**
The switch state is write in the memory **FD.SW.PitLim.State.**

3.2.2 SW Start Limiter (start BangBang Turbo).

The input is declared active to ground or active to Vref by the map **EE.SW.StartLim.LevelON**
- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.StartLim.ConfTime.**
The switch state is write in the memory **FD.SW.StartLim.State.**

3.2.3 SW Limiter.

The input is declared active to ground or active to Vref by the map **EE.SW.Limiter.LevelON**
- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Limiter.ConfTime.**
The switch state is write in the memory **FD.SW.Limiter.State.**

3.2.4 SW Upshift.

The input is declared active to ground or active to Vref by the map **EE.SW.Upshift.LevelON**
- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Upshift.ConfTime**
The switch state is write in the memory **FD.SW.Upshift.State.**

3.2.5 SW Double Map.

The input is declared active to ground or active to Vref by the map **EE.SW.DoubleMap.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.DoubleMap.ConfTime**
The switch state is write in the memory **FD.SW.DoubleMap.State.**

3.2.6 SW Fuel Level Init.

The input is declared active to ground or active to Vref by the map **EE.SW.Plein.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Plein.ConfTime**
The switch state is write in the memory **FD.SW.Plein.State.**

3.2.7 SW Perfo.

The input is declared active to ground or active to Vref by the map **EE.SW.Perfo.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Perfo.ConfTime**
The switch state is write in the memory **FD.SW.Perfo.State.**

3.2.8 SW Switch1.

The input is declared active to ground or active to Vref by the map **EE.SW.Sw1user.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Sw1user.ConfTime**
The switch state is write in the memory **FD.SW.Sw1user.State.**

3.2.9 SW Switch2.

The input is declared active to ground or active to Vref by the map **EE.SW.Sw2user.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Sw2user.ConfTime**
The switch state is write in the memory **FD.SW.Sw2user.State.**

3.2.10 SW Switch3.

The input is declared active to ground or active to Vref by the map **EE.SW.Sw3user.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Sw3user.ConfTime**
The switch state is write in the memory **FD.SW.Sw3user.State.**

3.2.11 SW Switch4.

The input is declared active to ground or active to Vref by the map **EE.SW.Sw4user.LevelON**

- 0 = ON to GND
- 1 = ON to VREF

If input selected is an analog input, the toggle level is internally set to 2.5 Volt.
A confirmation state time in mSec is configured in the map **EE.SW.Sw4user.ConfTime**
The switch state is write in the memory **FD.SW.Sw4user.State.**

3.3 Frequency Inputs

3.3.1 Engine Speed.

3.3.1.1 Engine Speed Derivative.

A derivative of engine speed is calculated and filtered every ½ engine rotation by the following formula:

$$\frac{dRPM}{dt} = \frac{dRPM}{dt} + ((RPM_n - RPM_{n-1}) * \frac{RPM_n}{30} - \frac{dRPM}{dt}) * EE.Ign.KfiltInfo$$

Where:

- ***RPM$_n$***, engine speed calculated in the last ½ engine rotation.
- ***RPM$_{n-1}$***, engine speed calculated in the previous ½ engine rotation.
- ***dRPM / dt,*** derivative of engine speed in RPM/second.
- ***KFILT_INFO*** no units (between 0 and 1).

3.3.1.2 Breakpoints.

3.3.1.2.1 Number of breakpoints for engine speed (long tables).

The table **EE.CfgU.NbkpReg** defines the number of breakpoints for tables function of the RPM requiring an important precision. The maximal value is 36.

3.3.1.2.2 Breakpoint Table (long).

The table **EE.Bkps.Regime** define the breakpoints (quantity defined above) expressed is in RPM and is used for addressing maps as a function of engine speed.

3.3.1.2.3 Number of breakpoints for the engine speed (short tables).

The table **EE.CfgU.NbkpRegShort** defines the number of breakpoints for tables function of the RPM requiring only a reduced precision. The maximal value is 24.

3.3.1.2.4 Breakpoint Table (short).

The table **EE.Bkps.RegShort** define the breakpoints (quantity defined above) expressed in RPM is used for addressing maps as a function of engine speed.

3.3.2 N_TURBO Speed.

3.3.2.1 N-Turbo Analogic Input configuration.

3.3.2.1.1 Diagnostic.

3.3.2.1.1.1 Validity Criteria

The acquired value of the N_turbo input, is first tested against maximum (**EE.TboBf.SAnaNturbo_Max**) and minimum (**EE.TboBf.SAnaNturbo_Min**) to verify its validity.

3.3.2.1.1.2 Default Mode

Should the value be outside these limits, a fault is signalled in the relevant diagnostic indicator and:

$$\boxed{\mathbf{N\_Turbo = EE.TboBf.NturboParDefaut (unit : 100 tr / mn)}}$$

3.3.2.1.2 Linearisation.

During normal operation the acquired value is linearised using the following formula:

$$\boxed{\mathbf{FD.Tbo.N\_Turbo = (N\_Turbo.Corr - EE.TboBf.Zero\_Nturbo) * 100 * EE.TboBf.Gain\_Nturbo}}$$

Where:

- ***FD.Tbo.N_Turbo*** in 10 Tr/mn.
- ***N_turbo.Corr, EE.TboBf.zero_Nturbo*** in Volt.
- ***EE.TboBf.gain_Nturbo*** in Tr/mn / mVolt.

3.3.2.2 N-Turbo Frequency Input configuration.

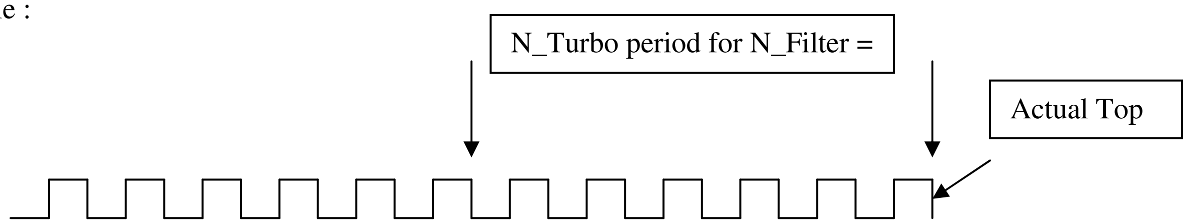
3.3.2.2.1 Top Number.

The top number from the flywheel is in the map **EE.TboBf.Nrev**. If a hardware prescaler is used between the sensor and the ECU , it's not the top number seen by the sensor but the top number seen by the ECU.

3.3.2.2.2 Filtering.

The Rpm Turbo calculation is issued from the period between **EE.TboBf.NFilter** top sample

Example :



3.3.2.3 N-Turbo Diagnostic

The value N_turbo, is tested, to verify its validity, against :

maximum value (**EE.TboBf.Nturbo_Max)**
minimum value (**EE.TboBf.Nturbo_Min)** only if engine rpm is greater than (**EE.TboBf.Reg_Nturbo_Min)** and the water temperature is greater than (**EE.TboBf.Teau_Nturbo_Min)**
Maximum absolute delta Nturbo (**EE.TboBf.D_Nturbo_Max)**