Trionic 7 Documentation

Dilemma, typeset in LaTeXby Jonathan Jogenfors

June 21, 2016

Todo list

Chapter 1

Introduction

This document is intended for Saab fanatics and engineers who want to start understanding the Saab Trionic 7 motor management system. It will give as much information as possible about the technical part of the system. The only limitation will be the knowledge of the author. In short the content of this document will enable you to understand Trionic 7 better and give you hands-on information about altering the maps it uses. Prerequisites are minor electronics and computer knowledge and of course some understanding of how a turbo charged engine works. Throughout the document the T7Suite software will be referenced. This software will enable you to really “get into” the Trionic. The T7Suite software can be downloaded from the T7Suite website.

Contents

Chapter 2

Hardware

The Trionic 7 is built around a Motorola MC68332 (CPU32) microcontroller. This is a 32-bit controller that handles the entire motor management including fuel injection, ignition timing and boost pressure control. The processor has a vast 4 MB (512 kB) flash memory for fetching program code and maps. This flash memory consisting of a AM29F400BT-90SI (AMD) holds the program memory. There is a coprocessor from Philips, a P83C592FHA/019. This is a 8-bit 8051-based microcontroller with a CAN (Controller Area Network) module. As the CAN physical line driver there is an Intel AN82527 (same family as used in Trionic 5). RAM memory is done by two 32 kbit SRAM chips (U62H256S1K). There is also a special component that would appear to be a barometric pressure sensor.

2.1 Integrated Circuit List

The table below lists almost all integrated circuits on the board. This is just to give you an idea on what to expect.

2.2 Block Schematic Diagram

2.3 DI Cassette

The DI cartridge has a trigger input for firing the four individual sparkplugs. These are triggered by signals from the ECU on pin 9, 10, 11 and 12 which are generated in the power driver IC CA3236 on the Trionic PCB (topside, 16 pin DIL housing). Internally these four pins are connected as shown in the table and the image below.

Partnumber	Function	Usage	# on board
TC55257DFI-85L	SRAM	(working memory)	1
16233970	Microcontroller	Main 32-bit CPU	1
PC83C592	Microcontroller with CAN contr.	8-bit coprocessor	1
AM29F400	Flash memory	4 Mbit	1
51862	DA converter		1
AN82527	CAN controller CAN line driver	1	
16238669 0H11	Pressure sensor?		1

Table 2.1

Figure 2.1

DI cartridge pin	ECU pinnumber	CA3236 pinnumber	Description
2	7	1 (OUT A)	Trigger cylinder 1
3	8	3 (OUT B)	Trigger cylinder 2
4	67	6 (OUT C)	Trigger cylinder 3
5	68	8 (OUT D)	Trigger cylinder 4

Table 2.2

Chapter 3

Flashing

The Trionic 7 ECU binary images uses several checksums to verify integrity. Most of them have been easy to figure out, but one of them is so complicated, that it seems to been done to deter map changing. There are still some unknowns, that would be nice to figure out. For example, I've discovered some binaries don't seem to have all four checksums. And of course there could be more checksums that have gone unnoticed. Two of the checksums are in the end of the binary and the other two are scattered in the code. The latter ones can be found by using pattern searching. Again the calculations are pretty simple, and even the harder checksum is easy to implement. Big thanks to solving these things goes to Tomi and General Failure.

3.1 Checksums

First of all, there are four different checksums. They have been given names by Tomi: FB checksum, F2 checksum, Misc checksum, and Area 70000 checksum.

Who is this?

3.1.1 FB and F2 checksums

The first two checksums, FB and F2, can be found at the end of the binary. This end area has been called the file header (footer would be more logical). See also chapter Trionic 7 file header. The F2 checksum is not present in all binaries, so be aware of this. Finding the two other checksums is more of a challenge.

Add reference

3.1.2 Misc checksum

The Misc checksum resides inline within the code. It is usually found somewhere between 0x02000 and 0x05000. The checksum address can be found by pattern searching the bin file using a set of hex values along with mask bits. If a mask bit is not set, the corresponding hex value does not have to match. Here are the hex values: 0x48, 0xE7, 0x00, 0x3C, 0x24, 0x7C, 0x00, 0xF0, 0x00, 0x00, 0x26, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x28, 0x7C, 0x00, 0xF0, 0x00, 0x00, 0x2A, 0x7C and the mask is 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1.

So, we are searching the binary for a string of bytes beginning with 0x48, 0xE7, 0x00, 0x3C.... Then we mask out the bytes that change from binary to binary. When we've located this pattern, we know where to start. Now we start primitively disassembling the code. We search for byte patterns 0x48, 0x6D, 0x48, 0x78, 0x48, 0x79, 0x2A, 0x7C and 0xB0, 0xB9. The three first patterns reveal addresses and lengths of checksum areas. There are 15 checksums areas from which the Misc checksum is calculated. The 0x2A, 0x7C pattern gives a base address for the 0x48, 0x6D addresses.

Bear with me. These 0x2A, 0x7C addresses are summed with the base address to make the actual address. This way the address is only 2 bytes long. On the 0x48, 0x79 addresses it's 4 bytes long without any base address. And the 0x48, 0x78 pattern gives 2 bytes which correspond with the length of the checksum area. Finally the 0xB0, 0xB9 pattern is followed by 4 bytes to the address of the Misc checksum.

3.1.3 Area 70000 checksum

This checksum refers to an area in the region of 0x70000. Like the Misc checksum, there is no clean way of finding out the length of the area along with the checksum address. Using pattern searching with this also has results. There are binaries that are incompatible with this approach, so this requires some fixing in the future. The pattern is 0x20, 0x3c, 0x00, 0x00, 0x11, 0x52, 0x2F, 0x00, 0x20, 0x3C, 0x00, 0x00, 0x09, 0xD0, 0x2F, 0x00, 0x20, 0x3C, 0x00, 0x00, 0x00, 0xCC, 0xD0, 0x9F and the mask is 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1

After finding that pattern, the masked addresses are summed together. In the pattern, the original addresses are 0x00001152, 0x000009D0 and 0x000000CC. Summing these gives the Area 70000 length of 0x1BEE. At the same time this is the address where to find the Area 70000 checksum.

3.2 Computing the Checksums

The FB checksum shares the same calculation method as the Misc and Area 70000 checksums. It's simply a sum of the bytes from the checksum area. Four bytes are made into a 32 bit value and summed with the next 32-bit value. This goes on until there are fewer than 4 bytes left. The last 1, 2 or 3 bytes are then individually summed together with the checksum.

The Misc checksum is a sum of the individual checksums calculated from 15 areas. The checksum calculation used is the same as with the FB checksum.

Once you have found the length of Area 70000, you can calculate the checksum by using the function described in section 3.1.1. The start address is 0x70000. Notice that your binary might not have this area present.

Chapter 4

Firmware

Once you are done with dumping the flash contents and you want to do more than only alter variables and maps you can start analyzing the binary. This is a difficult task because there are a lot of different firmware versions, stock ones – maybe different per MY and tuned ones that differ for every manufacturer and stage. In every case the code can be disassembled using a `6833x` disassembler like the one in `IDA Pro`. There are scripts available to automatically disassemble the code and make it more readable by replacing addresses by variable names that are extracted from the symbol table inside the binary.

Chapter 5

Symbol table

Each Trionic 7 firmware file contains a symbol table describing data structures in the program. The major problem is that from some point in time SAAB started to compress the symbol tables in the binary file. Probably just to save space in the flash memory but it has made tuning a little harder. We actually need these symbol names because they tell us what a certain memory location means. For unpacked binaries these symbols can be extracted together with their corresponding memory addresses (ROM and RAM).

While examining the symbol table you can see that the separator is 0x00. In contrast to T5 where symbol and SRAM addresses reside in the same table, we now only find the symbol name. In another table in the binary we can find flash addresses and lengths in the same sequence as the symboltable.

5.1 Finding start of symbol table:

Search binary for string the first sequence of 15 zeros.

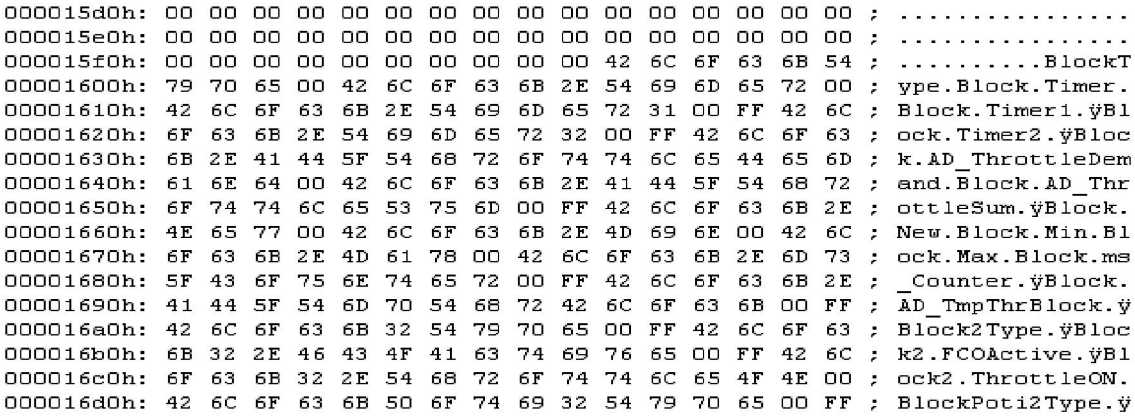
5.2 Finding start of address lookup table:

Search binary for 20 00 00 00 XX YY 00 F0 where XX YY is the index of the first symbol found in the symbollist.

To save you the time to lookup all addresses manually the T7Suite application will extract all symbol information in one run. Symbol name, flash address and length will be displayed all together.

This image (6) will give you an idea of what the symbol table should look like once it has been

link



```
000015d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000015e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000015f0h: 00 00 00 00 00 00 00 00 00 00 42 6C 6F 63 6B 54 ; ..........BlockT
00001600h: 79 70 65 00 42 6C 6F 63 6B 2E 54 69 6D 65 72 00 ; ype.Block.Timer.
00001610h: 42 6C 6F 63 6B 2E 54 69 6D 65 72 31 00 FF 42 6C ; Block.Timer1.ÿBl
00001620h: 6F 63 6B 2E 54 69 6D 65 72 32 00 FF 42 6C 6F 63 ; ock.Timer2.ÿBloc
00001630h: 6B 2E 41 44 5F 54 68 72 6F 74 74 6C 65 44 65 6D ; k.AD_ThrottleDem
00001640h: 61 6E 64 00 42 6C 6F 63 6B 2E 41 44 5F 54 68 72 ; and.Block.AD_Thr
00001650h: 6F 74 74 6C 65 53 75 6D 00 FF 42 6C 6F 63 6B 2E ; ottleSum.ÿBlock.
00001660h: 4E 65 77 00 42 6C 6F 63 6B 2E 4D 69 6E 00 42 6C ; New.Block.Min.Bl
00001670h: 6F 63 6B 2E 4D 61 78 00 42 6C 6F 63 6B 2E 6D 73 ; ock.Max.Block.ms
00001680h: 5F 43 6F 75 6E 74 65 72 00 FF 42 6C 6F 63 6B 2E ; _Counter.ÿBlock.
00001690h: 41 44 5F 54 6D 70 54 68 72 42 6C 6F 63 6B 00 FF ; AD_TmpThrBlock.ÿ
000016a0h: 42 6C 6F 63 6B 32 54 79 70 65 00 FF 42 6C 6F 63 ; Block2Type.ÿBloc
000016b0h: 6B 32 2E 46 43 4F 41 63 74 69 76 65 00 FF 42 6C ; k2.FCOActive.ÿBl
000016c0h: 6F 63 6B 32 2E 54 68 72 6F 74 74 6C 65 4F 4E 00 ; ock2.ThrottleON.
000016d0h: 42 6C 6F 63 6B 50 6F 74 69 32 54 79 70 65 00 FF ; BlockPoti2Type.ÿ
```

Figure 5.1

Figure 5.2

Figure 5.3

extracted. See appendix I for a complete list of known symbols.

When the user double clicks one of the symbols that has a flash address attached to it, T7Suite will display the corresponding symbol in a viewer. This viewer will display the data in table form was well as in graphical form.

Chapter 6

Mapping

A lot of maps in Trionic 7 are not only made up of a piece of raw data. It also includes x-axis and y-axis information. T7Suite will automatically display all known axis information when a map is opened. In Trionic 7 most symbols have an English name (Trionic 5 has lots of Swedish names) that explains lots about its function. Also, the symbols are categorized by name, which makes browsing the symbols much easier. All torque calibration symbols start with "TorqueCal.". T7Suite groups all symbols by their respective category by default.

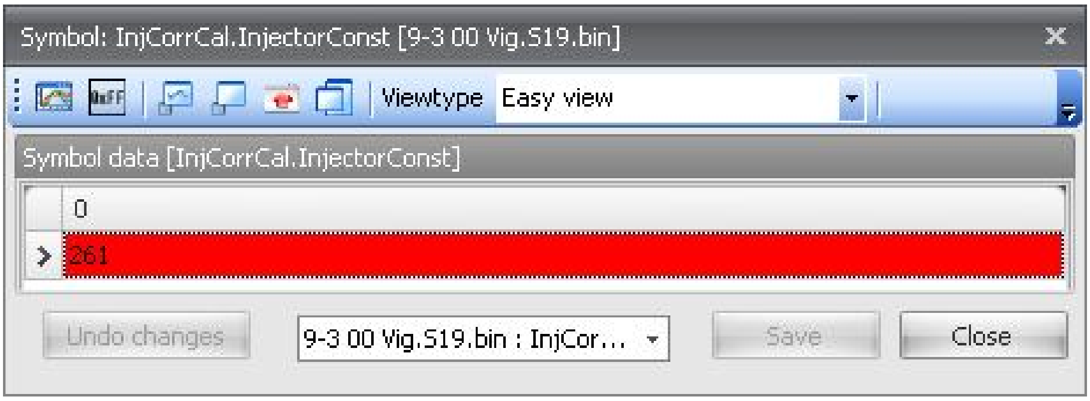
6.1 Fuel

Fuel calculation in Trionic 7 is based on the Airmass entering the engine. In rough steps this seems to be the flow of the calculation:

1. Basic calculation of fuel quantity per combustion. The air mass per combustion in mg is converted to fuel mass in mg by dividing by the AFR. This is dependent on fuel type:

$$\text{mGasoline} = \frac{\text{mAir}}{14.7}, \qquad \text{mE85} = \frac{\text{mAir}}{9.765} \tag{6.1}$$

2. In case of a cold engine, shortly after starting, rapid load changes, knocking or high loads, the fuel mass is multiplied by a compensation factor.
3. The closed loop value is used as a multiplier.
4. Correction for purge. Multiply by the value for purge adaptation. The value is sent to box 5
5. Long term fuel trim. The fuel mass is multiplied by the multiplicative adaptation value.
6. Additive adaptation. The additive adaptation value is added to the fuel mass.
7. Starting fuel quantity. If the engine has not yet started, starting fuel is selected.
8. The fuel quantity per combustion is the amount of petrol to be supplied to the engine.
9. Computation of injector opening duration. Converts fuel mass to opening time from a lookup table.
10. Injection takes place twice per combustion until the camshaft position has been found. Injection duration is divided by two.
11. Battery voltage correction. Injector opening duration is corrected by a voltage-dependent factor that is added to the duration.


Symbol: InjCorrCal.InjectorConst [9-3 00 Vig.S19.bin]

Viewtype Easy view

Symbol data [InjCorrCal.InjectorConst]

0

261

Figure 6.1: InjCorrCal.InjectorConstant

12. If fuel cut is active, the computed injector duration is replaced by a fuel cut value.

After this computation, the current injector is opened for the computed duration of time at a determined crank shaft angle. We will now look at the maps that control the above computation.

The basic fuel quantity is calculated based on air mass (measured from the air mass flow sensor) and injector constant (map `InjCorrCal.InjectorConstant`). If the engine has reached operating temperature, the standard map `BFuelCal.Map` is used. If not, the map is switched out for the cold start map `BFuelCal.StartMap`.

The areas calibrated to be run in closed loop have a value of 1.00 ± 0.02. Note that the high load part of the maps ramp up to enrich the mixture. Therefore, adjust the closed loop part of the map first.

To do this, turn closed loop mode off and drive with a wideband lambda sensor. What you want to achieve is an AFR of 14.7 while driving around under light loads (i.e. 1.00 ± 0.02 as per above). What Trionic 7 then does is to compute an injector duration. If the measured AFR does not achieve the correct value, the map must be altered. For example, if Trionic 7 computes a duration of 5 ms, a map value of 1.1 results in a duration of $5 * 1.1 = 5.5$ ms which means more fuel and a richer mixture. Conversely, a value below 1.1 results in a leaner mixture.

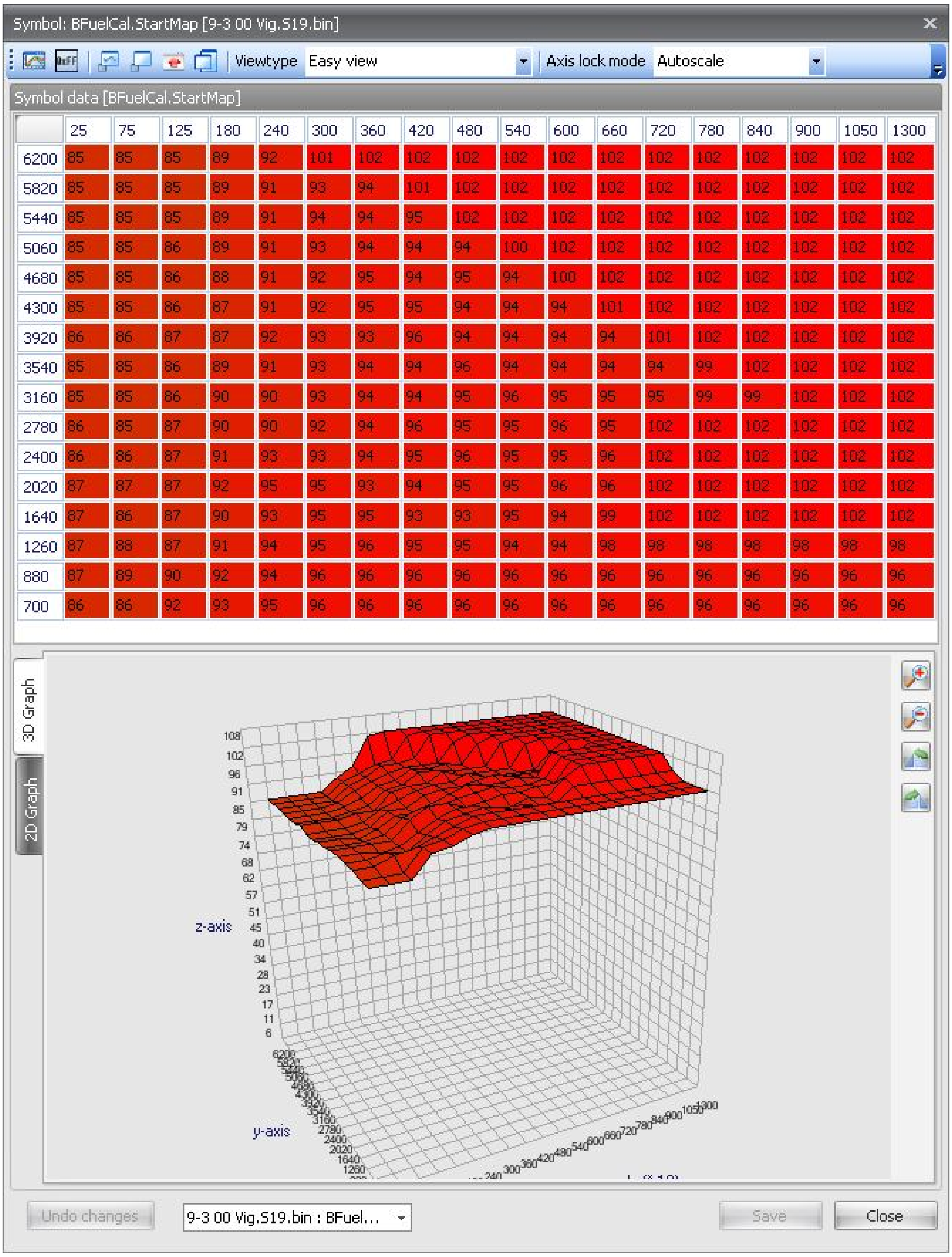
The idea is to get the closed loop area correct first. This will stop any negative adaption once the tune is finished and allowed to run in closed loop again. If the closed loop area of your fuel map is too rich it will negatively adapt over a long period of time. This will have the effect of leaning your air-fuel ratios across the board.

For example, let's look at a tune with an AFR that is correct in closed loop. This means that the feedback from the O_2 sensor is enough to correct the combustion. This can hide a condition where the mixture is too rich by 13 % and the O_2 sensor reduces the fuel mass by 13 %. Then, when full power is applied, Trionic 7 enters open loop mode and the wideband lambda sensor measures an AFR of 12.5.

If this condition remains, Trionic 7 will adjust the long term fuel trim (multiplicative adaption, `Amul`) to be -13 % after a number of weeks. This makes the fuel calculation reduce the fuel injection by 13 % over the whole range, including full power. Here's the danger. At the next full power run, there will be 13 % less fuel than ideal, resulting in an AFR of 14.0. The engine now runs lean and *can fail very quickly, causing massive internal engine damage*.

Now back to fuel mapping. To set closed loop area of fuel map, monitor the AFR in this light load area and if its wrong after adjusting for large injectors, start by just adjusting the Injector constant up and down accordingly instead of altering the closed loop area of the `BfuelCalMap`. This will affect the whole map instead of one point. By doing it this way you can almost get AFR spot on in closed loop area just by a few goes at adjusting the injector constant.

It can be found in `InjCorrCal.InjectorConst`, the value represents the injectors flow in mg of fuel (not capacity or cc's as injectors are normally rated in). The injector constant is a calculation factor



Symbol: BFuelCal.StartMap [9-3 00 Vig.S19.bin]

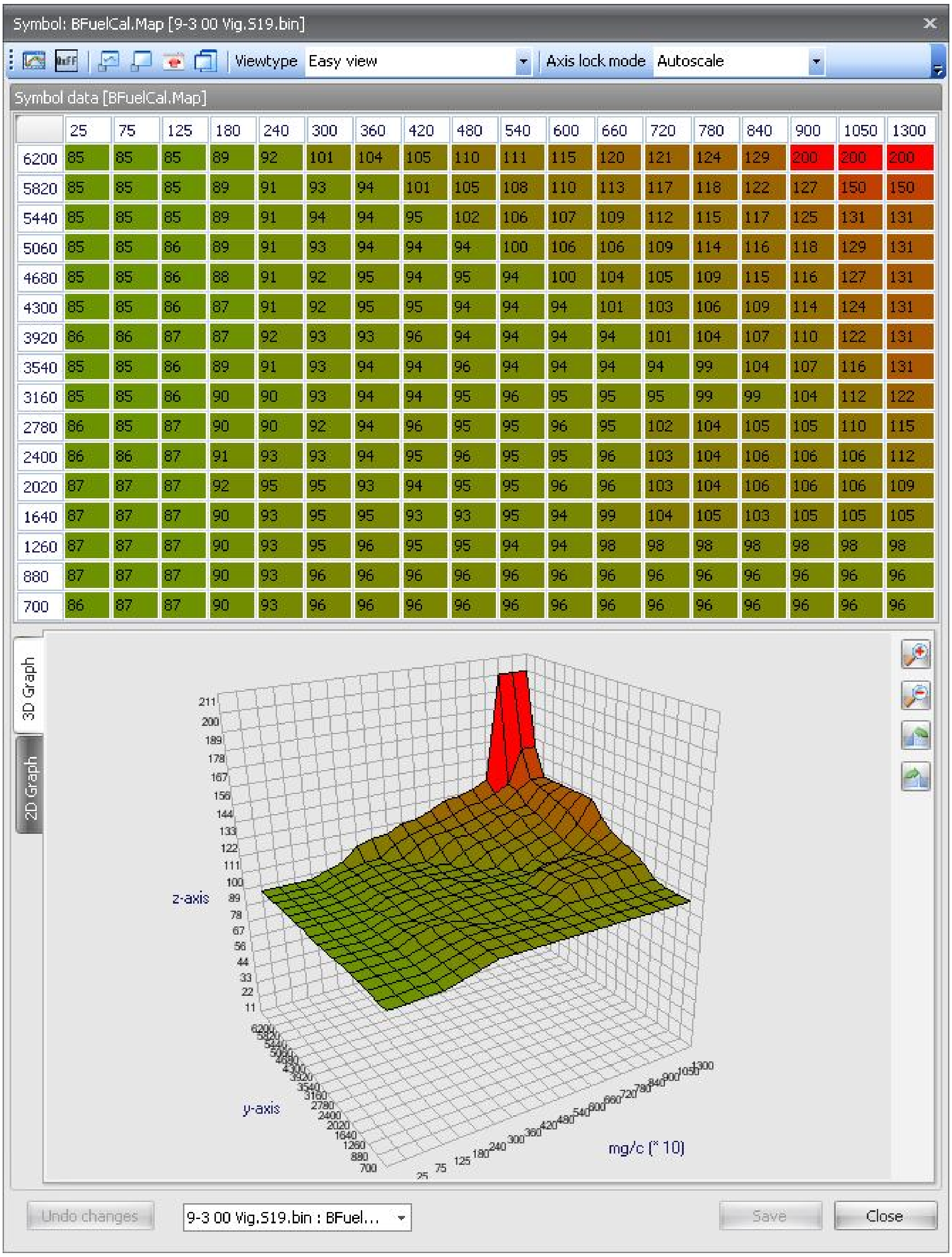
Viewtype: Easy view

Axis lock mode: Autoscale

Symbol data [BFuelCal.StartMap]

	25	75	125	180	240	300	360	420	480	540	600	660	720	780	840	900	1050	1300
6200	85	85	85	89	92	101	102	102	102	102	102	102	102	102	102	102	102	102
5820	85	85	85	89	91	93	94	101	102	102	102	102	102	102	102	102	102	102
5440	85	85	85	89	91	94	94	95	102	102	102	102	102	102	102	102	102	102
5060	85	85	86	89	91	93	94	94	94	100	102	102	102	102	102	102	102	102
4680	85	85	86	88	91	92	95	94	95	94	100	102	102	102	102	102	102	102
4300	85	85	86	87	91	92	95	95	94	94	94	101	102	102	102	102	102	102
3920	86	86	87	87	92	93	93	96	94	94	94	94	101	102	102	102	102	102
3540	85	85	86	89	91	93	94	94	96	94	94	94	94	99	102	102	102	102
3160	85	85	86	90	90	93	94	94	95	96	95	95	95	99	99	102	102	102
2780	86	85	87	90	90	92	94	96	95	95	96	95	102	102	102	102	102	102
2400	86	86	87	91	93	93	94	95	96	95	95	96	102	102	102	102	102	102
2020	87	87	87	92	95	95	93	94	95	95	96	96	102	102	102	102	102	102
1640	87	86	87	90	93	95	95	93	93	95	94	99	102	102	102	102	102	102
1260	87	88	87	91	94	95	96	95	95	94	94	98	98	98	98	98	98	98
880	87	89	90	92	94	96	96	96	96	96	96	96	96	96	96	96	96	96
700	86	86	92	93	95	96	96	96	96	96	96	96	96	96	96	96	96	96

Figure 6.2: BFuelCal.StartMap



Symbol: BFuelCal.Map [9-3 00 Vig.S19.bin]

Viewtype Easy view

Axis lock mode Autoscale

Symbol data [BFuelCal.Map]

	25	75	125	180	240	300	360	420	480	540	600	660	720	780	840	900	1050	1300
6200	85	85	85	89	92	101	104	105	110	111	115	120	121	124	129	200	200	200
5820	85	85	85	89	91	93	94	101	105	108	110	113	117	118	122	127	150	150
5440	85	85	85	89	91	94	94	95	102	106	107	109	112	115	117	125	131	131
5060	85	85	86	89	91	93	94	94	94	100	106	106	109	114	116	118	129	131
4680	85	85	86	88	91	92	95	94	95	94	100	104	105	109	115	116	127	131
4300	85	85	86	87	91	92	95	95	94	94	94	101	103	106	109	114	124	131
3920	86	86	87	87	92	93	93	96	94	94	94	94	101	104	107	110	122	131
3540	85	85	86	89	91	93	94	94	96	94	94	94	94	99	104	107	116	131
3160	85	85	86	90	90	93	94	94	95	96	95	95	95	99	99	104	112	122
2780	86	85	87	90	90	92	94	96	95	95	96	95	102	104	105	105	110	115
2400	86	86	87	91	93	93	94	95	96	95	95	96	103	104	106	106	106	112
2020	87	87	87	92	95	95	93	94	95	95	96	96	103	104	106	106	106	109
1640	87	87	87	90	93	95	95	93	93	95	94	99	104	105	103	105	105	105
1260	87	87	87	90	93	95	96	95	95	94	94	98	98	98	98	98	98	98
880	87	87	87	90	93	96	96	96	96	96	96	96	96	96	96	96	96	96
700	86	87	87	90	93	96	96	96	96	96	96	96	96	96	96	96	96	96

Figure 6.3: BFuelCal.FuelMap

Figure 6.4: Battery correction values for injector latency

Figure 6.5: Water temperature correction

used by Trionic 7 to calculate injection time.

Once closed loop area is done the high load area can be mapped. If its too lean just increase the values in the relative column relating to what site of the map you are running in. Once your high load areas are done, activate closed loop again so you can see how it all runs. Monitor fuel adaptations and AFR etc.

One thing to note is how quickly it drops into open loop under full throttle. By going into open loop the O2 sensor is "masked" where the ECU listens to what its saying but ignores it, this allows AFR to go beyond 14.7 and injection correction is directly taken from the fuel map we just adjusted allowing much enrichment to cool charge etc.

To alter open loop enrichment you can change at what Airmass flow Trionic 7 switches to open loop in `LambdaCal.MaxLoadNormTab`. Also, open loop entry (so, leaving closed loop situation) has a delay attached to it. This way, short overruns of the maximum load will not immediately result in leaving closed loop. Stock bins often have this set to 2000 milliseconds which seems quite long. If you want to ECU to leave closed loop faster after overrunning the load limit, just decrease the time in `LambdaCal.TimeOpenLoop`.

1. Idling speed ignition timing With idle speed control active, the timing is adjusted to stabilize idle engine speed. The value is sent to box 3

2. Normal ignition timing When idle speed control is inactive, the ignition timing is read from a load and engine speed depending matrix. The value from the matrix is optimized for lowest fuel consumption (best engine torque) and sent to box 3

3. Selection of ignition timing One of the ignition timing calculation is selected depending on which function is active. The value is sent to box 6

4. Catalytic converter heating timing In order to heat up the catalytic converter as fast as possible after

Figure 6.6: Fuel injection correction map for knock conditions

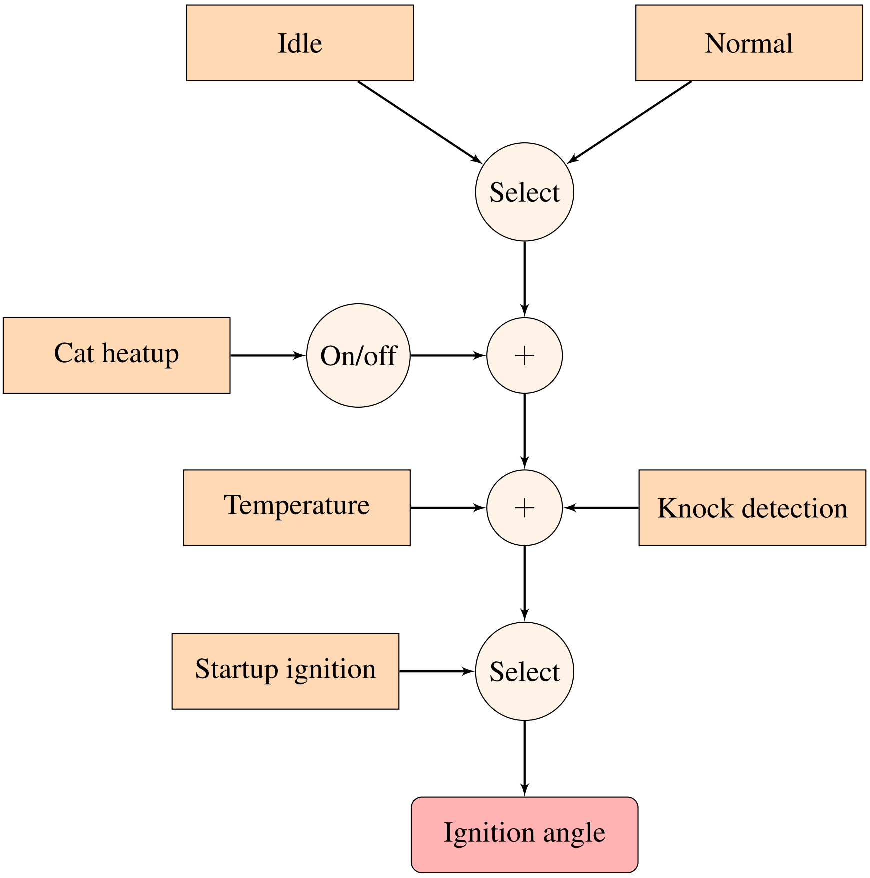


Figure 6.7: Flowchart for ignition calculation

Figure 6.8: ignition chart

start, the ignition will be retarded. This is a compensation matrix that is added to the value in box 3. The matrix is dependent on load and engine speed

5. Engagement of catalytic converter heating timing The function is active when coolant temperature is above -10 degrees Celsius and below +64 degrees Celsius
6. Total The value from box 5 is added to the value of box 3
7. Compensation The ignition timing is corrected depending on engine coolant temperature and intake air temperature. The value is sent to box 6.
8. Knock control If knocking occurs, a timing retardation will be calculated. The value is sent to box 6
9. Total The compensation angle and knock retardation are totalled to give the current ignition timing. The value is sent to box 7
10. Selection of ignition timing Starting ignition timing is selected when the engine has not been started. The value is sent to box 9
11. Starting ignition timing Starting ignition timing is selected when the engine has not yet been started. The value is sent to box 9
12. Activate relevant trigger At the calculated crankshaft angle, the microprocessor controls the transistor for the trigger that is next in firing order

6.1.1 DI Cassette

The direct ignition (DI) cassette is mounted on the valve cover on top of the spark plugs. The DI cassette houses four ignition coils/transformers whose secondary coil is directly connected to the spark plugs. The ignition cassette is electrically supplied with battery voltage from the main relay (B+) and is grounded in an earth point. When the main relay is activated the battery voltage is transformed to 400 V DC which is stored in a capacitor. The 400 V voltage is connected to one of the poles of the primary coil in the four spark coils. Connected to the DI cassette there are four triggering lines (from the Trionic ECU, pin 9 (cyl. 1), pin 10 (cyl. 2), pin 11 (cyl. 3) and pin 12 (cyl. 4)). When the ECU is grounding pin 9, the primary coil for the first cylinder is grounded (via the B+ intake of the DI cassette) and 400 V is transformed up to a maximum of 40 kV in the secondary coil for cyl. 1. The same procedure is used for controlling the ignition on the rest of the cylinders.

Figure 6.9: DI Cassette

Figure 6.10: idlecontrol

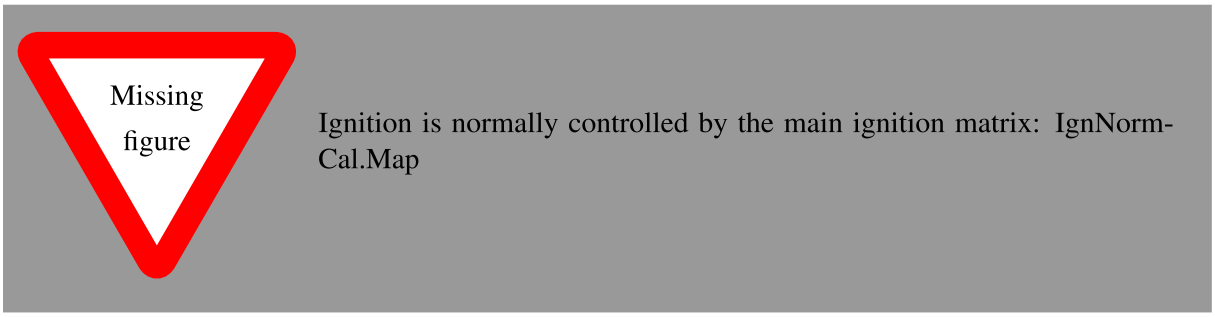

Missing
figure
Ignition is normally controlled by the main ignition matrix: IgnNormCal.Map

Figure 6.11: Ignition is normally controlled by the main ignition matrix: IgnNormCal.Map

6.1.2 Idle control

6.2 Torque

Trionic 7 is a torque/Airmass request system instead of a boost request system like Trionic 5 is. The basic procedure for the Airmass controller is like in the table below.

1. Driver request The control module reads pedal potentiometer 1 and converts the voltage to Airmass per combustion ($\mathrm{mg\,combustion}^{-1}$). The value is sent to box 3

2. Cruise control request When cruise control is active, the air mass per combustion required to maintain the set speed is calculated. The value is sent to box 3

3. Select highest value The control module selects the highest of the two values (box 1 or box 2). The value is sent to box 5

4. Engine torque limitation The maximum permissible air mass per combustion varies depending on the engine type. During operation, the maximum permissible $\mathrm{mg\,combustion}^{-1}$ must also be limited to protect the engine, gearbox, brakes and turbo

5. Select lowest value The control module selects the lowest value and sends it to box 8

6. Compensation request When the AC compressor is on, and when the heated rear window or radiator fan is on, the $\mathrm{mg\,combustion}^{-1}$ required to compensate for the increased load is calculated. The value is sent to box 8

7. Other air request The control module calculates the $\mathrm{mg\,combustion}^{-1}$ required for idle speed control. The value is sent to box 8

8. Totalling values The control module totals all the values. The total is sent to box 9

9. Total requested $\mathrm{mg\,combustion}^{-1}$

10. Total Airmass request

11. Throttle control The requested $\mathrm{mg\,combustion}^{-1}$ is converted to requested voltage for throttle position sensor 1. The charge air pressure and intake air temp are used to correct this conversion. The throttle motor rotates the throttle until the current voltage for throttle position sensor 1 corresponds with the requested voltage

12. Current $\mathrm{mg\,combustion}^{-1}$ The requested mgc is also compared with the current mgc (MAF reading). If needed the requested voltage for throttle position sensor 1 is finely adjusted

13. Turbo control If $\mathrm{mg\,combustion}^{-1}$ is too high for throttle alone the turbo control will take over. The excess is converted to a PWM which controls the charge air control valve. The absolute pressure sensor is used to correct the conversion

14. Current $\mathrm{mg\,combustion}^{-1}$ The requested mgc is compared to current mgc and the charge air control vale PWM is finely adjusted if required

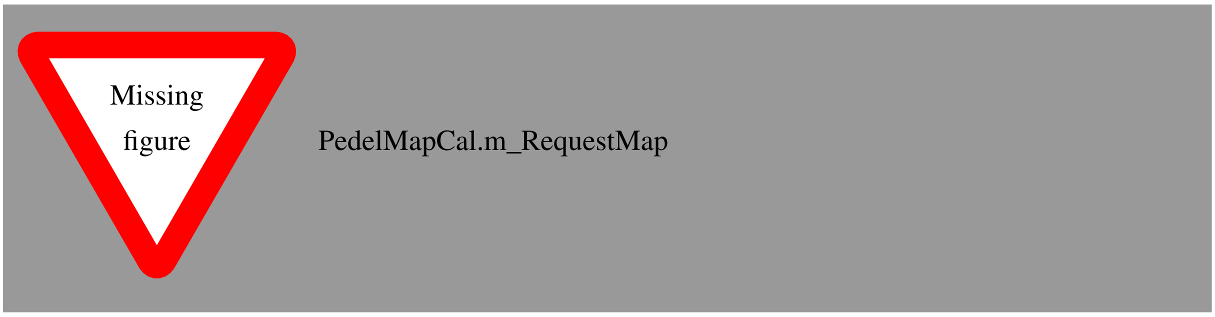


Figure 6.12: PedelMapCal.m_RequestMap

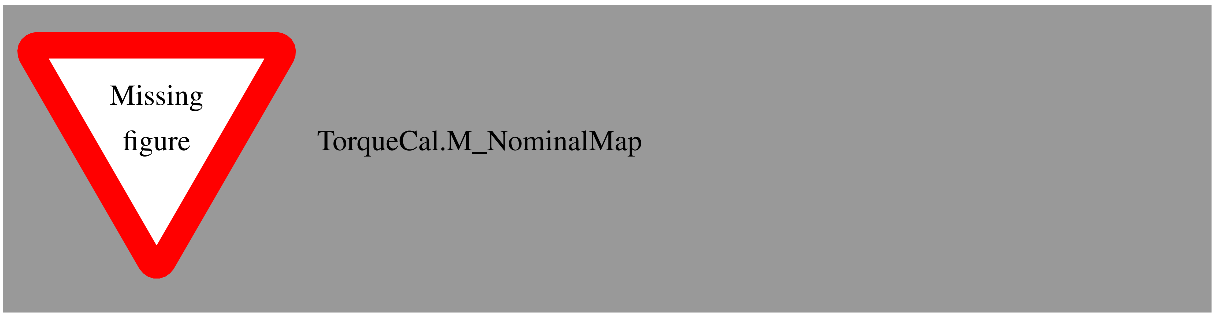


Figure 6.13: TorqueCal.M_NominalMap

6.2.1 Torque request

When the driver (or cruise control for that matter) presses the accelerator pedal this is interpreted as a certan airmass request from the system. This value is fetched from the `PedelMapCal.m_RequestMap` shown below.

The table holds Airmass values for each position of the accelerator pedal and each rpm site. Trionic now looks up the estimated engine output (torque) based on Airmass and rpm. This is done through map `TorqueCal.M_NominalMap` as shown next.

6.3 Second lambda sensor

In the years Trionic 7 was shipped on cars, several things changed in these cars setups. One of the major changes was the introduction of the second lambda (oxygen, O2) sensor that is placed after the catalyst to ensure the catalyst is working properly. If you want to run software from a double lambda sensor car in a single lambda car, you have to make some changes in the settings. This information is courtesy of L4staero.

6.3.1 Turning off second lambda sensor

You can use this procedure in cars having only one lambda sensor and in cars having two lambda sensors, but with a missing catalyst.

Map	Value	Description
LambdaCal.ST_AdapEnable	0	Second lambda sensor disabled
LambdaCal.ST_AdapEnable	1	Second lambda sensor enabled

Table 6.1

Map	Value	Description
O2HeatPostCal.I_LowLim	0	
CatDiagCal.LoadLo	20	Second lambda sensor disabled
CatDiagCal.LoadHi	30	
O2HeatPostCal.I_LowLim	230	
CatDiagCal.LoadLo	140	Second lambda sensor enabled
CatDiagCal.LoadHi	425	

Table 6.2

6.3.2 Alternative solution to turning off second lambda

Change low limit on `O2heaterPostCal.I_LowLim` to 0 mA to disable sensor heater error, and change `CatDiagCal.LoadHi` and `CatDiagCal.LoadLo` to values never seen normally, like 30 and 20.

6.4 Calibration of OBD2 and LEV EVAP systems

If we want to run a file that was developed for OBD2 or a LEV car in an earlier car we run into problems because the early car is missing a second catalyst, a tank pressure sensor and a purge canister behind the fuel tank. If you have an early B205E/L engine you simply couldn't run a later B205R software version in it because it would through CEL's for the missing hardware. We need to make changes to the file before we can run in on an earlier car (e.g. switch of the control of the new hardware).

- `OBDCal.OBD2Enabled` This is self explanatory, if car is OBD2 put value at 1 if it's not OBD2 put value at 0.

  On this point in later bins(compressed) there is EOBDEnable which is always on in EC2000 EU files and LOBDEnable which is always on in EC2000 RW files. File type is shown in firmware information under engine type.

- `OBDCal.EnableOBD2Limit` As above but its a 4 byte value. If done in Hex value for a OBD2 car is 00000001 and for non-OBD2 car is 00000000. As shown in T7suite is 2 values. OBD2 cars top value is 1 and bottom value is 0. In non OBD2 car both values are 0.

- `OBDCal.evapEquipmentExist` If car is equipped with a canister at rear of tank and a tank pressure sensor value will be 1. If neither exist value should be set to 0.

6.4.1 Info on LEV (Low Emission Vehicle)

For example take a 2001 Saab 9-3 Aero (or SE as called in USA) equipped with a B205R, Saab's decision to take all "R" engines and clean them so to speak by developing new emission systems for them leaves them with some differences to their low level engine relatives. The term LEV(Low emission vehicle) in this sense refers to Saab's decision to add a second catalytic converter and a tank pressure sensor and a large purge canister behind fuel tank, as well as adding a 2nd oxy to monitor condition of first cat.

Identifier	Length	Description
0x91	0x09	Ecuid.vehicleidnr
0x94	0x07	Ecuid.ecuhardwversnr
0x95	0x0C	Ecuid.ecusoftwnr
0x97	0x1E	Ecuid.ecusoftwversnr
0x9A	0x04	Ecuid.softwaredate
0x9C	0x04	variable name table crc (not really sure)
0x9B	0x04	Symboltable (packed table with symbol names)
0xF2	0x04	F2 checksum
0xFB	0x04	Romchecksum.piareachecksum
0xFC	0x04	Romchecksum.BottomOffFlash
0xFD	0x04	RomChecksumType
0xFE	0x04	Romchecksum.TopOffFlash
0xFA	0x05	Lastmodifiedby
0x92	0x0F	Ecuid.partnralphacode (IMMO)
0x93	0x07	Ecuid.ecuhardwnr
0xF8	0x02	?
0xF7	0x02	?
0xF6	0x02	?
0xF5	0x02	?
0x90	0x11	Ecuid.scaletable (VIN)
0x99	0x06	Ecuid.testerserialnr
0x98	0x0D	Ecuid.enginetype
0xF9	0x01	Romchecksum.Error

Table 6.3

6.5 Footer information

If we look at the footer in the binary (last page in hex viewer) we see a set of reversed strings. Each of these strings contains an identifier. These identifiers have a hardcoded meaning.

Figure 6.14: footer

Chapter 7

Tuning Trionic 7

7.1 Tuning with T7Suite

To get the ECU to produce more engine output, several parameters (maps) have to altered. This chapter will give you a general idea on what to change – and why – for getting to an approximate stage II equivalent. The example is a 9-3 B205R.

7.1.1 AirCtrlCal.m_MaxAirTab

Airmass value from controller where area map has reached max-area and there is no point to increase the I-part. Resolution is 1 mg combustion^{-1}

7.1.2 AirCtrlCal.m_MaxAirE85Tab

(if running on E85) Same as above for E85

7.1.3 BoostCal.I_LimTab

Load limit tab. to enable the I Part of boost regulator. If the load request from Airmass master is above this value plus the hysteresis is the I Part enabled and the throttle closed loop is disabled. If the load request from Airmass master is below this value is the I Part disabled and the throttle is allowed to run in closed loop.

Figure 7.1: AirCtrlCal.m_MaxAirTab

Figure 7.2: MaxAirTab

Figure 7.3: ILimTab

7.1.4 BoostCal.P_LimTab

Load limit tab. to enable the P Part of boost regulator. If the load request from Airmass master is above this value plus the hysteresis is the P Part enabled. If the load request from Airmass master is below this value is the P Part disabled.

7.1.5 BoostCal.RegMap

Main constant matrix. Resolution is 0.1 %.

Figure 7.4: Plimtab

Figure 7.5: RegMap

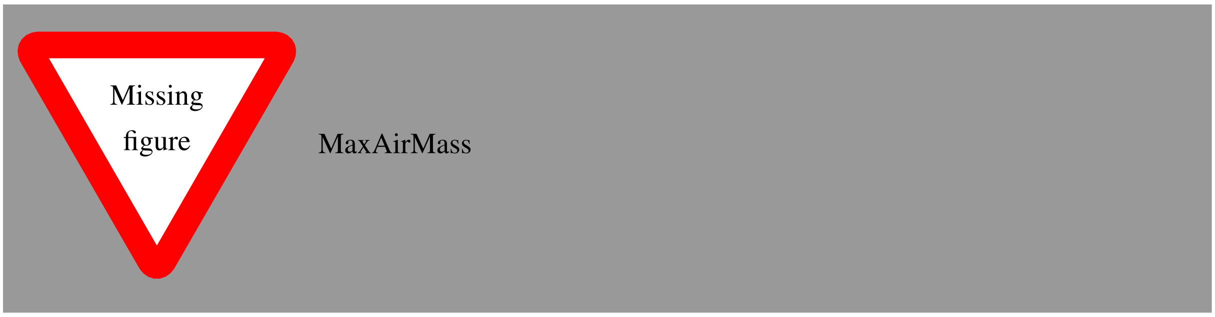


Figure 7.6: MaxAirMass

7.1.6 BstKnkCal.MaxAirmass

(divide by 3.1 for approx torque, ignition, airtemp etc affect this!) Map for max allowed Airmass for manual gearbox, `m_nHigh`. Resolution is 1 mg combustion^{-1}.

7.1.7 BstKnkCal.MaxAirmassAu

Map for max allowed Airmass for automatic gearbox, m_nHigh. Resolution is 1 mg combustion^{-1}.

7.1.8 FCutCal.m_AirInletLimit

If the "MAF.m_AirInletFuel" is higher than this limit during m_AirInletTime will the fuelcut be activated (pressure guard).

7.1.9 IgnE85Cal.fi_AbsMap

(if you want to change the ignition) Ignition map for E85 fuel. Resolution is 0.1 degrees.

7.1.10 IgnNormCal.Map

(if you want to change the ignition) Normal ignition map. Resolution is 0.1 degrees.

7.1.11 MapChkCal.CheckSum

(automatically updated in between every map change with T7suite!)

Figure 7.7: AirInletLimit

Figure 7.8: AbsMap

7.1.12 MaxVehicCal.v_MaxSpeed

Speed limiter.

7.1.13 PedalMapCal.m_RequestMap

Requested Airmass from the driver as a function of rpm and accelerator pedal position. Resolution is 1 mg combustion^{-1}.

7.1.14 TorqueCal.M_ManGearLim

Maximum engine torque limit for each gear in the manual gearbox. Resolution is 1 Nm.

Figure 7.9: RequestMap

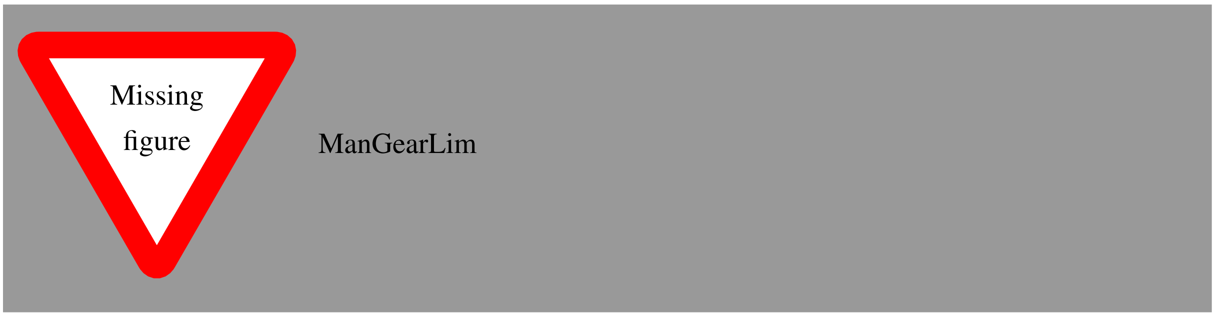


Figure 7.10: ManGearLim

Figure 7.11: AirTorqMap

7.1.15 TorqueCal.m_AirTorqMap

(This is where all torque limiters take their data from and therefore needs to be "fooled" if you are running 400nm+ or an automatic!) Data-matrix for nominal Airmass. Engine speed and torque are used as support points. The value in the matrix + friction Airmass (idle Airmass) will create the pointed torque at the pointed engine speed. Resolution is 1 mg combustion^{-1}. axis to the above map: TorqueCal.m_AirXSP

7.1.16 TorqueCal.M_EngMaxTab

Data-table for maximum engine out put torque for manual cars. Resolution is 1 Nm.

Figure 7.12: EngMaxTab

Figure 7.13: EngMaxAutTab

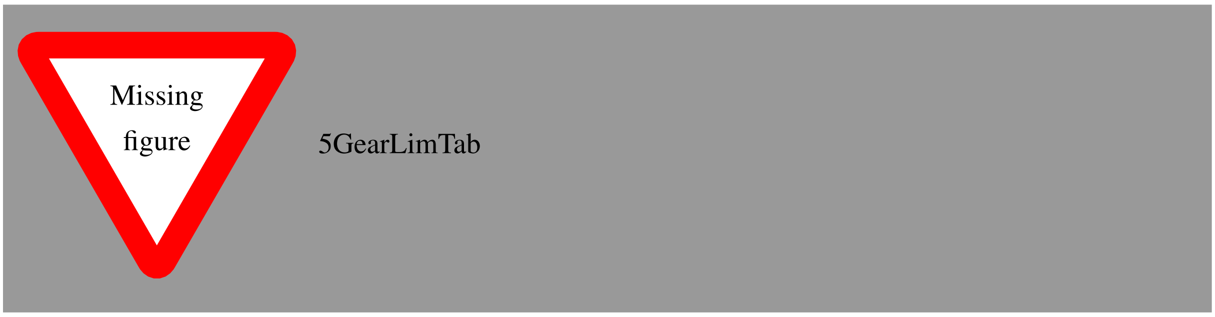


Figure 7.14: 5GearLimTab

7.1.17 TorqueCal.M_EngMaxAutTab

Data-table for maximum engine output torque for automatic cars. Resolution is 1 Nm.

7.1.18 TorqueCal.M_5GearLimTab

Data-table for maximum engine output torque for manual cars on fifth gear. Resolution is 1 Nm.

7.1.19 TorqueCal.M_EngMaxE85Tab

(if running on E85) Data-table for maximum engine output torque when running on E85. Resolution is 1 Nm.

7.1.20 TorqueCal.m_PedYSP

Air mass support points for (Calc) X_AccPedalMap. Resolution is 1 mg combustion^{-1}.

7.2 Tuning Boost calibration

This map holds percentages (0.1% accurate) of how much air should be passed to the return hose of the boost control value. The higher the value, to more air is bled off and the less the wastegate will open (and thus, the more air the turbo will be spooling). As you can see, the more Airmass is requested (x axis) the more the wastegate is held shut and thus, the more Airmass the turbo will be providing. If we want more Airmass from the turbo, we need to keep the wastegate shut longer and thus we have to enter higher numbers on the right side of the table.

Figure 7.15: PedYSP

Figure 7.16: BoostCal

7.3 Altering Airmass limiter

To be able to flow more air though the engine that is allowed in the stock configuration we will have to modify the Airmass limiter tables as well. Note that there are two different ones, one for manual gearbox and one for automatic gearbox. This example will only show the manual gearbox table (`BstKnkCal.MaxAirmass`) but for automatic cars `BstKnkCal.MaxAirMassAu` needs to be changed.

As you can see, the maximum amount of Airmass allowed is approximately $970\,\mathrm{mg\,combustion^{-1}}$. We need to change the table so that it will allow more Airmass. In this case we just up the table with $25\,\%$ with the math functions in T7Suite. NOTE: Please do not simply turn off this limiter by setting it way higher than the actually intended level because it is an important limiter to provide engine safety.

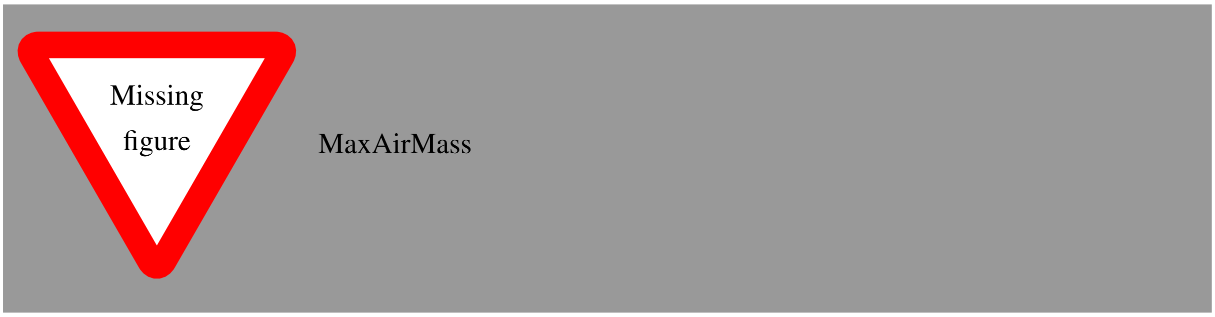


Figure 7.17: MaxAirMass

Figure 7.18: FuelCut

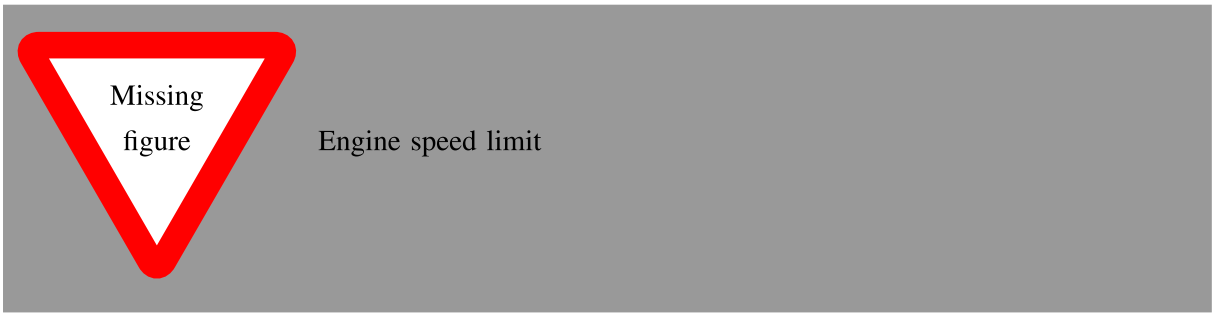


Figure 7.19: Engine speed limit

7.4 Altering fuelcut

Then there's the fuelcut function to worry about. We need to increase the limit of what the fuel cut function will accept to prevent it from shutting of fuel too early. NOTE: Please do not simply turn off this limiter by setting it way higher as the actually intended level because it is an important limiter to provide engine safety.

7.5 Engine speed limiter

To prevent the system to reduce Airmass above engine speeds that are still acceptable we need to change `MaxSpdCal.n_EngLimAir` as well. Y axis values are engine temperature (coolant). Please note that 200 rpm above this limit, the fuel cut mechanism will become active!

7.6 Vehicle speed limiter

An option is to increase the vehicle speed limiter as well. In this stock binary the vehicle speed is limited to $240\,\mathrm{km\,h^{-1}}$. We can change it to – for example $280\,\mathrm{km\,h^{-1}}$.

7.7 Airmass request

To get more from the engine than in the stock configuration we need to actually request more Airmass for a certain pedal position and rpm site. This can be done through `PedalMapCal.m_RequestMap`. Because

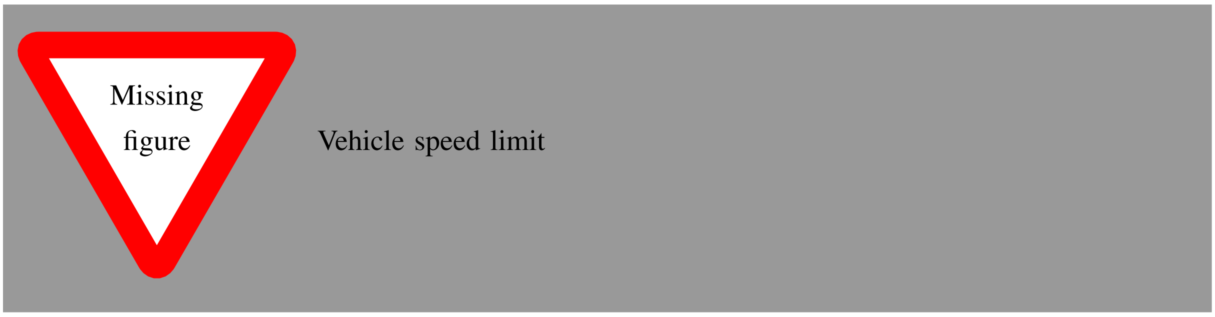


Figure 7.20: Vehicle speed limit

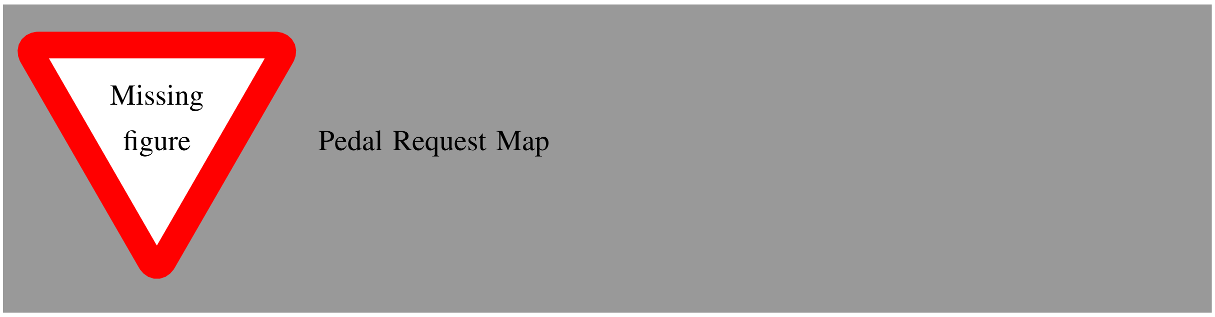


Figure 7.21: Pedal Request Map

we want more power at wide open throttle (from the drivers perspective) we need to increase the Airmass request at pedal positions in the high percentage range (top of the table).

As you can see we increased the top two rows so that a maximum of 1350 mg combustion^{-1} will be requested. In addition we need to alter the y axis support point for the pedal map that lets Trionic lookup a pedal position for a given Airmass. This map is called `TorqueCal.m_PedYSP`. This axis map should support the maximum Airmass we're requesting in the `TorqueCal.m_Requestmap`, so in our case we need to modify the map to match the 1350 mg combustion^{-1} we are requesting as a maximum.

The map that uses this axis is called `TorqueCal.x_AccPedalMap`. It it shown below with the altered axis values for clarification.

Figure 7.22: PedYSP

Figure 7.23: AccPedalMap

Figure 7.24: EngMaxAutTab

7.8 Torque limiter

To prevent to system to reduce Airmass above a certain engine output, the torque limiter needs to be increased according to expected engine output.

Torque limiter for E85 fuel Torque limiter for manual gearbox in higher revs Torque limiter in 5th gear Overboost table Maximum Airmass for I-part of PID controller (`AirCtrlCal.m_MaxAirTab`)

`TorqueCal.m_AirTorqMap`. This is where all torque limiters take their data from and therefore needs to be "fooled" if you are running more than 400 N m on an automatic!

Data-matrix for nominal Airmass. Engine speed and torque are used as support points. The value in the matrix + friction Airmass (idle Airmass) will create the pointed torque at the pointed engine speed. Resolution is 1 mg combustion^{-1}.

Finally, we're all done!

Figure 7.25: EngMaxE85Tab

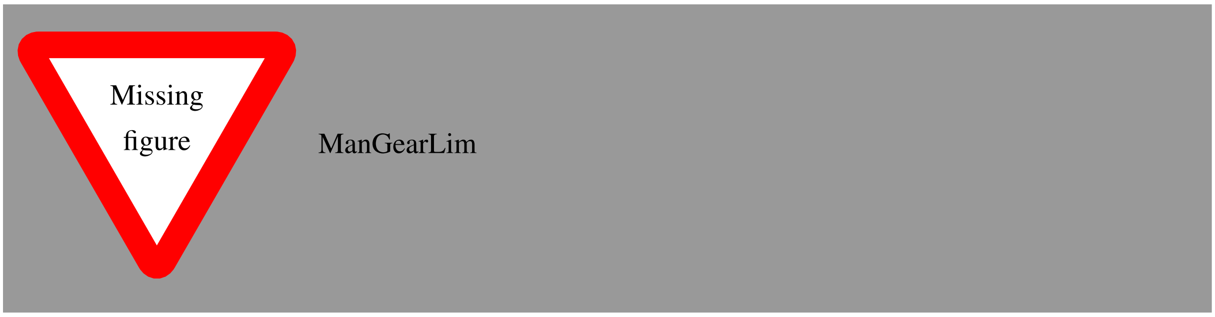


Figure 7.26: ManGearLim

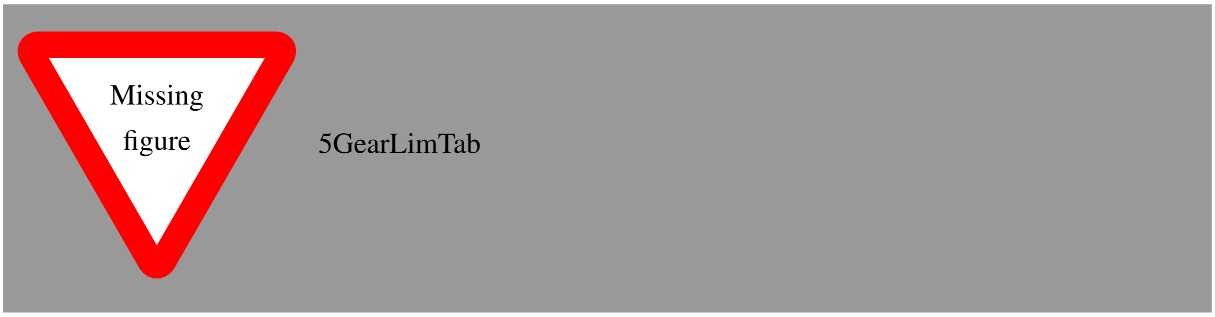


Figure 7.27: 5GearLimTab

Figure 7.28: Overboost

Figure 7.29: MaxAirTab

Figure 7.30: AirTorqMap

Chapter 8

Automatic transmission specifics

In automatic Trionic 7 cars the TCM (Traction Control Module) sends a torque limit over can to the ECU (Trionic). This means – theoretically - you cannot achieve more torque than the torque limit the TCM dictates. The only known way around this at present time is to make Trionic THINK it's not making that much torque, so we have to fool the ECU into thinking it is still below the torque limit set by the TCM.

There are different TCM limits depending on year, engine type, gearbox etc. A MY01-AERO AUT has a 330 N m limiter while a 5 speed automatic gearbox has a 350 N m limit. To fool the ECU into thinking it is making less torque is to rescale the x-axis for `TorqueCal.mAirTorqMap` which is `TorqueCal.M_EngXSP`. The top value in this list must be no more than the TCM limit.

In this case the top three rows have been altered to keep the calculated torque below 330 N m.

This means that when requesting 330 N m you will actually get 400 N m, 320 N m will get you 350 N m and so on. The torque limiters in `TorqueCal.M_EngMaxAutTab` must be scaled with this in mind. In this case 400 N m between 2780 rpm to 3920 rpm. Values in between you need to recalculate, at 4300rpm the user wanted 390Nm, (400=330, 350=320) means that 322=360, 324=370, 326=380, 328=390nm

If you want to use the same bin in manual cars all manual limiters must be calculated and set correctly!

NOTE: In Bio power bins TorqueCal.M_EngMaxE85Tab !

400	->	330Nm
350	->	320Nm
320	->	310Nm

Table 8.1

Figure 8.1: TorqueCal EngXSP

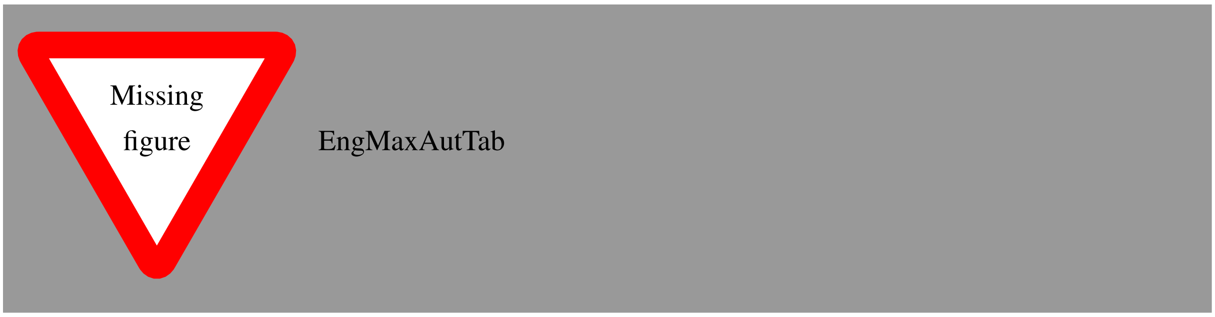


Figure 8.2: EngMaxAutTab

Figure 8.3: columns

Chapter 9

Using the tuning wizard

Emphasize th itations of tur wizard

T7Suite incorporates a tuning wizard. This wizard allows you to automatically alter the maps in the binary file to get it to a stage I equivalent file. The wizard can be activated by selecting `Tuning->Easy tune to stage 1` from the menu. A dialog will appear in which you can confirm that you want to tune the file to stage I. Once you've selected this the process will start. After a few seconds a report will appear showing all actions taken on your file.

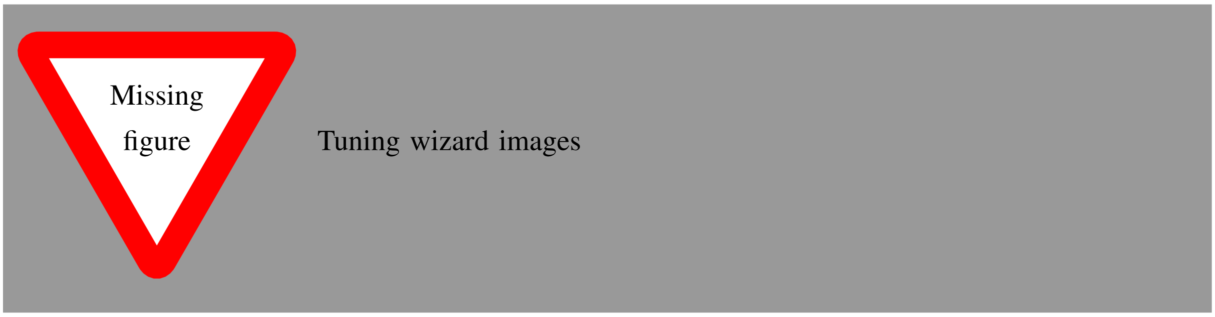


Figure 9.1: Tuning wizard images

Chapter 10

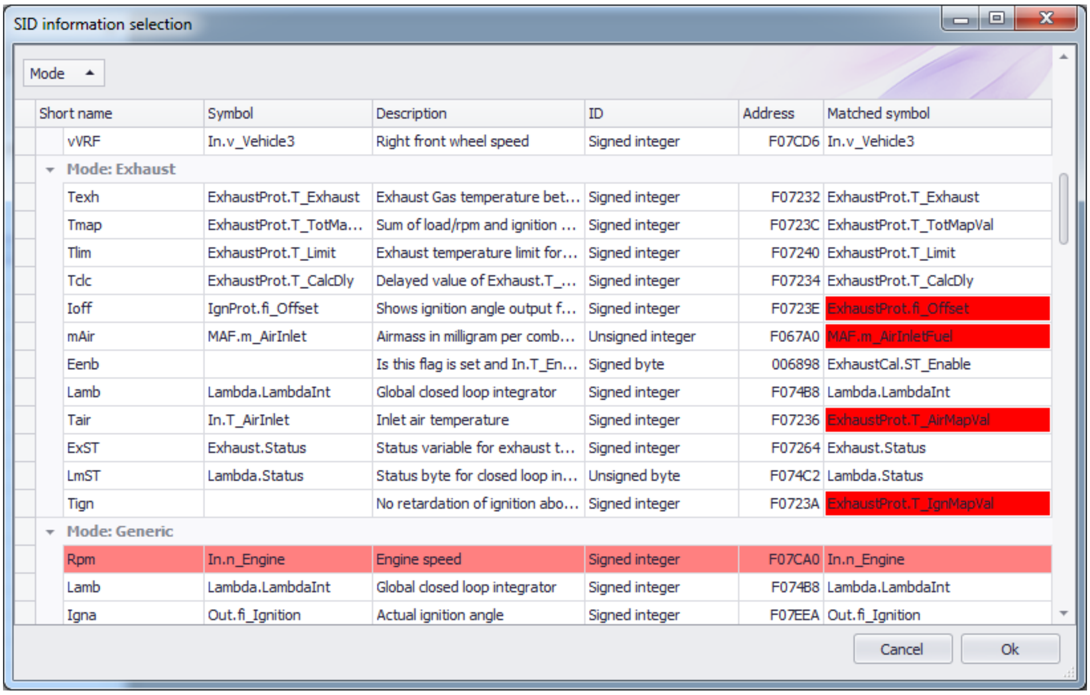
OpenSID values

T7Suite incorporates a function to allow visualization of information on the SID (System Information Display). This way you can view real-time information without utilizing the Canbus interface. You can select the variables you want the SID to display using the SID information selection option in T7Suite.

Also, the software should be opened to be able to view the selected data on the SID. This is done in the firmware information screen.

Some tips on how to use the SID information option:

- ECMStat.ST_ActiveAirDem shows the current Airmass limiter
- ECMStat.P_Engine shows calculated engine power (hp)
- ECMStat.AirFuelRatio shows calculated AFR
- ECMStat.p_Diff shows boost pressure (manifold – ambient) in 0.1 kPa units
- BstKnkProt.MapPointer shows the offset in 0.1 degrees for BstKnk.MaxAirmass (so, the ignition offset for knock)
- ExhaustCal.ST_Enable allows you to enable and disable the EGT algorithm. These algorithms are based on the stock engine and won't be properly calibrated for a stage 3+ setup.
- KnkDetAdap.KnkCntCyl first 2 bytes show cylinder 1 knock count, next 2 bytes shows cylinder 2 knock count etc.



Short name	Symbol	Description	ID	Address	Matched symbol
vVRF	In.v_Vehicle3	Right front wheel speed	Signed integer	F07CD6	In.v_Vehicle3
Mode: Exhaust					
Texh	ExhaustProt.T_Exhaust	Exhaust Gas temperature bet…	Signed integer	F07232	ExhaustProt.T_Exhaust
Tmap	ExhaustProt.T_TotMa…	Sum of load/rpm and ignition …	Signed integer	F0723C	ExhaustProt.T_TotMapVal
Tlim	ExhaustProt.T_Limit	Exhaust temperature limit for…	Signed integer	F07240	ExhaustProt.T_Limit
Tclc	ExhaustProt.T_CalcDly	Delayed value of Exhaust.T_…	Signed integer	F07234	ExhaustProt.T_CalcDly
Ioff	IgnProt.fi_Offset	Shows ignition angle output f…	Signed integer	F0723E	ExhaustProt.fi_Offset
mAir	MAF.m_AirInlet	Airmass in milligram per comb…	Unsigned integer	F067A0	MAF.m_AirInletFuel
Eenb		Is this flag is set and In.T_En…	Signed byte	006898	ExhaustCal.ST_Enable
Lamb	Lambda.LambdaInt	Global closed loop integrator	Signed integer	F074B8	Lambda.LambdaInt
Tair	In.T_AirInlet	Inlet air temperature	Signed integer	F07236	ExhaustProt.T_AirMapVal
ExST	Exhaust.Status	Status variable for exhaust t…	Signed integer	F07264	Exhaust.Status
LmST	Lambda.Status	Status byte for closed loop in…	Unsigned byte	F074C2	Lambda.Status
Tign		No retardation of ignition abo…	Signed integer	F0723A	ExhaustProt.T_IgnMapVal
Mode: Generic					
Rpm	In.n_Engine	Engine speed	Signed integer	F07CA0	In.n_Engine
Lamb	Lambda.LambdaInt	Global closed loop integrator	Signed integer	F074B8	Lambda.LambdaInt
Igna	Out.fi_Ignition	Actual ignition angle	Signed integer	F07EEA	Out.fi_Ignition

Figure 10.1: SID Information

Chapter 11

CAN bus interface

The most common way to interface with the Trionic 7 unit is via the CAN bus.

11.1 General information

The most frequently used interface for this is the Lawicel CANUSB interface[*]. This interface can convert CAN signals onto you USB port and vice versa. The interface has a USB port on one side – that connects to you computer – and an male `DB9` connector on the other side. This side connects to the CAN bus of the Trionic.

The Lawicel interface has the following pin out on the DB9 connector.

11.2 OBD2 socket pinout

On some models, the OBDII port enables you to connect to the I bus directly. On most models, you need to wire into the P-pus (preferably, because data transmission rates are tenfold of that on the Ibus) or into the I-bus directly. The P-bus can be found at the pins of the ECU (as described on the previous page), the I-bus can be found in a lot of places like the CD changer connector in the trunk (other spots are shown in the table below).

Section shoul
updated with
obd2 informa

11.3 Saab I-bus communication

Courtesy of Tomili and General Failure

The I-Bus is an internal bus (Instrumentation Bus) that connects together instruments such as the radio, ACC and the SID (Information Display). The I-bus is the non-critical bus which means that vehicle critical information is not sent over it. That is done through the P (powertrain) bus. The I-bus enables us to communicate with several devices in the car including the TWICE, DICE etc. Both the I and the P bus are CAN bus based communication buses which enables us to use a normal CAN adapter to interface with them.

[*] www.canusb.com
[†] Only on some models
[‡] Only on some models

CAN controller	`Intel AN825257`
Communication speed	615 $\mathrm{kbit\,s^{-1}}$

Table 11.1: CAN specifications for Trionic 7

Figure 11.1: DB9

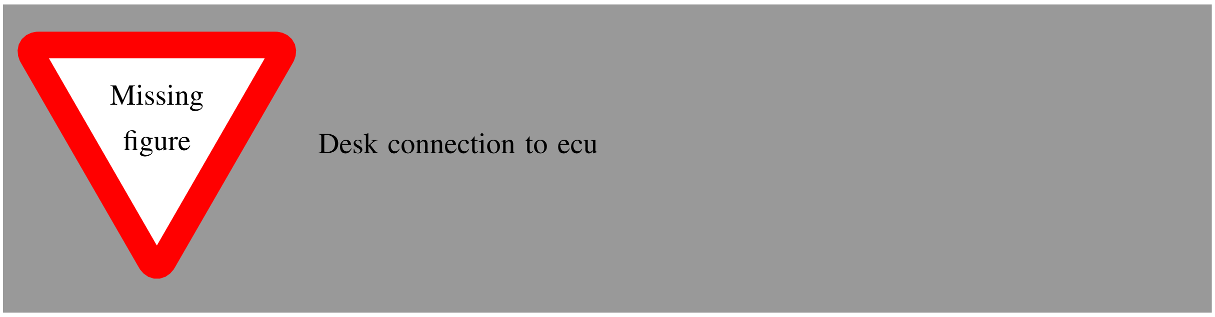


Figure 11.2: Desk connection to ecu

Pin number	Description
1	
2	J1850 Bus+
3	
4	Chassis ground
5	Signal ground
6	CAN High (J-2284)[†]
7	ISO 9141-2 K-line
8	
9	
10	J1850 Bus-
11	Airbag Controller (?)
12	ABS Controller (?)
13	
14	CAN Low (J-2284)[‡]
15	ISO 9141-2 L-line
16	Battery Power

Table 11.2

The I bus follows the ISO 11898-2 standard which is the high speed variant, although it only implements a maximum communication speed of 47,619 Kbit/s. Messages on a CAN bus are sent within CAN frames that include an identifier number, number of data bytes, the actual data (called the payload) and checksum (to verify it the data was transferred correctly). There are two CAN frame formats: the basic frame (referred to as CAN2.0A) and the extended frame (referred to as CAN2.0B). The basic frame has a 11-bit identifier field, when the extended frame has a 29-bit identifier which makes it possible to extend the number of message types that can be sent over the bus. The I bus uses two signal (just like any other CAN bus connection) which are called I+ and I-. These refer to CAN High and CAN Low signals in general CAN terminology. Details on I bus communication:

Frame type: CAN 2.0A (11-bit identifiers) Bus speed: 47,619 Kbit/s Timing register settings: BTR0 0xCB, BTR1 0x9A

As an example the engine speed is located as a 16-bit integer value on message ID 460h (hexadecimal) in the second and third bytes (I define first byte as the most significant byte). Speed is in the same message in fourth and fifth bytes. The speed value is multiplied by ten, so you have to divide the 16-bit integer by 10 to get the real speed value. Here's an example ID 460h message: 00 03 9C 00 2A 00 00 00 Engine rpm is 039Ch (hexadecimal format) = 924 (decimal format) = 942 RPM Speed is 002Ah = 42 = 4,2 km/h

Finish I-bus a
bus section

Chapter 12

General tuning advice

This chapter will describe some frequently made mistakes in handling the Trionic.

Ignition advance When altering the ignition tables keep in mind that more than 35 degree advance from TDC is not good.

Ignition retard When altering the ignition tables keep in mind that more than 5 degreen retarding from TDC is not good.

Ignition advance at WOT Try to keep the ignition advance at approximately 10 degrees BTDC

Recommended AFR Try to keep the AFR between 10.8 and 12.5 at WOT.

Recommended EGT Try to keep the exhaust gas temperature below 950 degrees C.

Determining the maximum boost request for a given turbo Check that the boost request level fits somehow to compressor map: http://www.squirrelpf.com/turbocalc/index.php In addition you can read appendix IV.

Internal link

Recommended maximum intake air temperature Up to 60 °C is good enough. If temperatures rise above 60 °C consider replacing your stock intercooler with an aluminium, cross-flow type. These are available from Speedparts, Abbott, ETS and others.

Chapter 13

Acknowledgements

The author would like to thank everyone on ecuproject for their help on getting all this information together. Special thanks go out to General Failure, J.K. Nilsson, Hook, Hma, Vigge, Mackan, Sandy Rus, JKB, L4staero and Steve Hayes.

The sections with technical information on checksums and Saab CAN bus were originally written by Tomi and General Failure.

Appendix A

BDM Technical Information

BDM stands for Background Debug Mode. This refers to the mode the Motorola microcontroller is forced into when activating the BDM interface. This mode enables us to hold the processor in the program execution and read and write data from and to the memory inside the microcontroller and the memory connected to it. In this way we can download and program the flash contents which gives us access to the binaries we like so much! The BDM software you need can be downloaded from http://www.xendus.se/bdm/bd32-122.zip.

A.1 Home build 2 chips design schema

An alternative to buying a BDM interface can be building one yourself. This chapter will hand you all information needed to buy the components needed and the schema to build the interface. The image below shows the schema for the 2 chip design. There is also a 5 chip design and a GAL based design but these are more difficult to build at home.

Table A.1 lists the components you need to build the BDM interface. Of course a soldering iron, PCB etc. are things that you also need.

A.2 BDM Pinout

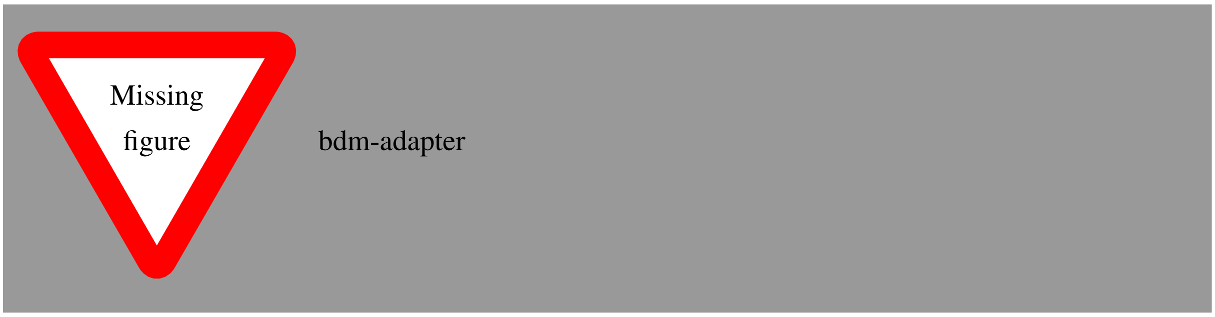


Figure A.1: bdm-adapter

Component	Amount	Description
74HC74	1	Dual JK Flip-Flop with Set and Reset
74HC132	1	Quad 2-input NAND Schmitt Trigger
Capacitor	0.1 uF	1
Capacitor	0.001 uF	1
Resistor	10 kohm	1
LPT cable + connector	1 meter	Smart is to get PCB type with normal LPT cable.
10 wire flat cable	20 cm	
10 pin female header for flat cable	1	

Table A.1: DIY BDM interface part list

Pin	Name	Description
1	DS	Data strobe from target MCU. Not used in current interface circuitry
2	BERR	Bus error input to target. Allows development system to force bus error when target MCU acce
3	VSS	Ground reference from target
4	BKPT/DSCLK	Breakpoint input to target in normal mode; development serial clock in BDM. Must be held lo
5	VSS	Ground reference from target
6	FREEZE	Freeze signal from target. High level indicates that target is in BDM
7	RESET	Reset signal to/from target. Must be held low to force hardware reset
8	IFETCH/DSI	Used to track instruction pipe in normal mode. Serial data input to target MCU in BDM
9	VCC	+5V supply from target.BDM interface circuit draws power from this supply and also monitors
10	IPIPE/DSO	Tracks instruction pipe in normal mode. Serial data output from target MCU in BDM

Table A.2

Appendix B

Understanding turbo compressor maps

Each turbo has its own characteristics. These are determined by the size of the turbine housing, the size of the compressor wheel, the size of the turbine blades and many more parameters. The most important identification of a turbocharger is by its compressor map. This is a graphical representation of its efficiency. In Saab Trionic 7 cars there are two commonly used turbo chargers; the Garrett GT17 for all models except for B205R and B235R. These latter engines are instead equipped with the Mitsubishi TD04-HL-15T (6cm2).

For an introduction to turbocharger terminology and concepts, please see [**GarrettTerminology**]. For information on basic flow calculation and how to choose between turbochargers, see [**TurboSelection**].

- Compressor and turbine wheels. The turbine wheel is the vaned wheel that is in the exhaust gases from the engine. It is propelled by the exhaust gases themselves. The turbine wheel is connected to the compressor wheel by an axle. So, the compressor wheel will spin together with the turbine wheel. The compressor wheel also is vaned and these vanes compress the air and force it into the intercooler.

- Wheel "trim". Trim is an area ratio used to describe both turbine and compressor wheels. Trim is calculated using the inducer and exducer diameters. As trim is increased, the wheel can support more air/gas flow.

- Compressor and turbine housing A/R. A/R describes a geometric property of all compressor and turbine housings. Increasing compressor A/R optimizes the performance for low boost applications. Changing turbine A/R has many effects. By going to a larger turbine A/R, the turbo comes up on boost at a higher engine speed, the flow capacity of the turbine is increased and less flow is wastegated, there is less engine backpressure, and engine volumetric efficiency is increased resulting in more overall power.

- Clipping. When an angle is machined on the turbine wheel exducer (outlet side), the wheel is said to be 'clipped'. Clipping causes a minor increase in the wheel's flow capability, however, it dramatically lowers the turbo efficiency. This reduction causes the turbo to come up on boost at a later engine speed (increased turbo lag). High performance applications should never use a clipped turbine wheel. All Garrett GT turbos use modern unclipped wheels.

- CFM = Cubic feet per minute.

- Lbs/minute = pounds (weight) per minute.

- M3/s = cubic meters per second.

- Corrected Airflow. Represents the corrected mass flow rate of air, taking into account air density (ambient temperature and pressure)

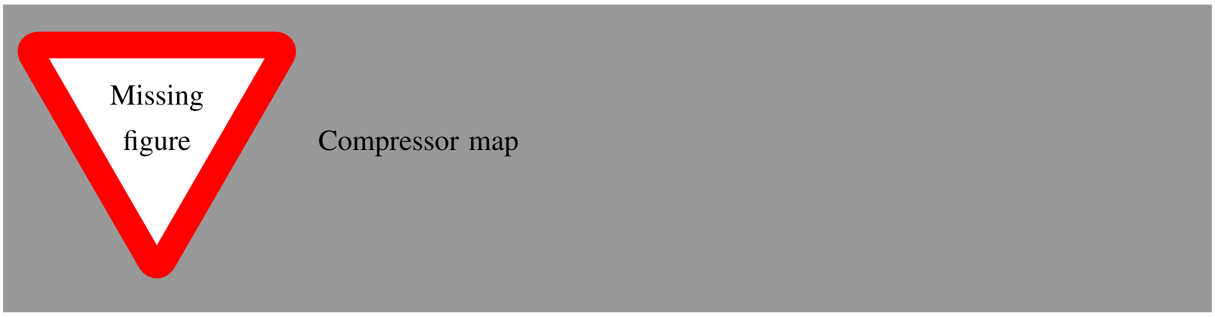


Figure B.1: Compressor map

Definition 1 The corrected airflow C_{corr} is defined as

$$C_{corr} = C_{air}\frac{13.95}{P}\sqrt{\frac{T_{air}+460}{545}} \tag{B.1}$$

ld only use

where T_{air} is the air temperature in Fahrenheit, P the barometric pressure in psi, and C_{air} the engine air consumption in lb/min.

Example 1 With $T_{air} = 60F$, $P = 14.7$ psi and $C_{air} = 50$ lb/min we get

$$C_{corr} = 50\frac{13.95}{14.7}\sqrt{\frac{60+460}{545}} = 46.3 \text{ lb/min} \tag{B.2}$$

- Pressure ratio. Ratio of absolute outlet pressure divided by absolute inlet pressure.

Definition 2 The pressure ratio $R_{pressure}$ is defined as

$$R_{pressure} = \frac{P_{boost}+P_{IC}+P_0}{P_0-P_{AF}} \tag{B.3}$$

where P_{boost} is the intake manifold pressure (boost) in psi, P_{IC} is the intercooler pressure drop in psi, DP_{AF} the air filter pressure drop in psi and P_0 is the air pressure at sea level in psi.

Example 2 With $P_{boost} = 12$ psi, $P_{IC} = 2$ psi, $P_{AF} = 0.5$ psi and $P_0 = 14.7$ psi we get

$$R_{pressure} = \frac{12+2+14.7}{14.7-0.5} = 2.02 \tag{B.4}$$

B.1 Reading a compressor map

This section contains a short description of reading compressor maps. For a more detailed analysis, see **isaac-lowry_turbo_2004** [**isaac-lowry_turbo_2004**]. A 2-dimensional pressure map looks like this.

The curved lines indicate the rotation speed (rpm) of the compressor wheel. In the sample map above these values are 45050, 69750, 84200 up to 125650 rpm. This is how fast a turbine wheel spins!! The elliptical circle means the compressor's efficiency area. It's marked by the percent sign. The horizontal axis is the amount of air before turbo, (1 m3/s = 2118.88 cfm, 10 lb/min = 144.718 cfm).

The vertical axis is the pressure ratio, the ratio of air pressure leaving to the turbo to air pressure entering the turbo. Pressure Ratio=The pressure at compressor exducer vs. the pressure at compressor inducer. In another word, the ratio of the pressure of the air after compression vs. the pressure before compression. As you can see, the pressure ratio depends on the ambient pressure. For example, at sea level, a turbo boosts 14.7psi. Ambient pressure is 14.7psi. That's 2 pressure ratio (PR) on the compressor map. Take that turbo to a higher elevation, the ambient pressure is less than 14.7psi. If the turbo still boosts 14.7psi, the pressure ratio would be higher. Now on the compressor map, you will see by moving up along a vertical line (to pump out the same cfm) and turbo efficiency has decreased as the elevation increases (PR increases). Simply put, turbos lose performance and become less efficient as elevation gets higher.

B.1.1 Choke area

The area to the right of the outer most elliptical circle is the least efficient area, the choke area. It means when the compressor reaches certain rpm, the air moved by the compressor wheel in the diffuser area of the compressor housing is moving at or past the speed of sound. When the air speed reach sonic speed, the amount of air flow increase is very small as compressor wheel rpm increases. In plain words, the compressor has reached its limit. You can try to pump more psi, have the wheel spin faster, but very little more air is pumped out the turbo compressor. You can see now, the compressor housing will need to properly match the compressor wheel. If you simply stuff a big wheel inside a small compressor housing, the diffuse area will be very small. This causes the air inside the housing to move at higher speed. That's why some of the so-called T28s which use a bigger compressor wheel inside the stock compressor housing does not produce good hp.

B.1.2 Compressor maximum flow

The max flow of a compressor is shown on the compressor map. On the map, look for the intersection of maximum compressor wheel speed (rpm) and the least compressor efficiency curve. Find that intersection. The horizontal coordinate is the max flow. The area to the right of maximum flow is the 'choke area'.

The vertical coordinate is the pressure ratio at which the compressor reaches that maximum flow. From this boost level, as the boost increases, very little air flow is increased. For example, if a compressor reaches its maximum flow at 2 PR or 1 atm pressure or 14.7psi, higher boost does not pump more air into the motor. But higher boost may be needed to increase the manifold pressure for the motor to flow more air. A 5 liter motor with this turbo needs 15psi of manifold pressure to flow a certain CFM. A 3 liter motor with the same turbo will need much higher manifold pressure to flow the same amount of air although that turbo's compressor does not flow more air past 14.7psi.

B.1.3 Compressor maximum pressure

On the map, find the top-most point on the graph. The vertical coordinate is the max pressure ratio. For example, 2.8 pressure ratio at sea level is 1.8 times the atmospheric pressure, 1.8x14.7psi=26.46 psi. Compressor max pressure is limited by compressor wheel speed. It's physically impossible to boost higher than this maximum pressure for one particular turbo. Plus the pressure drop in the intercooler system, the actual maximum boost reading from a boost gauge that's plugged into the intake manifolds maybe a few psi lower than this maximum pressure.

B.1.4 What the compressor map reads

Most manufactures rate their turbos at 1 bar (15 psi). That's 2 pressure ratio. On the map, draw a horizontal line from 2PR. When the line intersects the right-most elliptical circle, the corresponding number on the x-axis is the maximum cfm the turbo can flow at 1 bar. Use the TD04-15G's map for example, where the 2 PR line hits the right-most efficiency curve, it reads 428cfm as its flow rate at 15psi.

RPM	2.0L	2.3L
1000	35.3	40.6
2000	70.6	81.2
3000	105.9	121.8
4000	141.3	162.4
5000	176.6	203.1
6000	211.9	243.7
7000	247.2	284.3

Table B.1: Engine volumetric flow (H) in cfm as given by equation (B.5) for different rpm.

B.1.5 Comparing compressor maps

Well, compressor maps are really 3-dimensional maps. Any compressor map looks a hill/peak in 3 dimensions. Our compressor maps look like if you look at the hill directly from above vertically. The elliptical lines of elevations are the efficiency curves. Since in theory, we can always boost more and decrease turbo efficiency to get more cfm, let's set the same Pressure Ratio and compare turbos at the same efficiency curves.

As a rule of thumb, a large turbo will be better at making a lot of pressure but will spool slower than a small turbo. A small turbo will build boost fast but is less capable to make big boost pressure.

B.2 Selecting a different turbocharger

In order to put in a new turbocharger we first need some theory [**isaac-lowry_turbo_2004**].

warning

Definition 3 The engine volumetric flow (H) in cfm is defined as

$$H = \frac{D_{ci}}{1728} \times \frac{n}{2} \tag{B.5}$$

where D_{ci} is the engine displacement in cubic inches and n is the number of crankshaft revolutions per minute.

We can then find the airflow in lb/min by computing [**isaac-lowry_turbo_2004**].

$$N = \frac{29PH}{10.73T} \tag{B.6}$$

where P is the absolute pressure in psi and T is the absolute ambient temperature in Rankin.

nvert to met-

Tables B.1 and B.2The following tables apply to our Saab 2.0 and 2.3 liter engines. Since the amount of air to be flowed by the turbo is largest when RPM is at its top we will take the worst case scenario and get EVF @ 7000 RPM. We have to make an assumption on the ambient temperature which we will set at 20C. This is 68 Fahrenheit which is (460 + 68) = 528 Rankin. Now we can calculate the airflow of the engine in lb/min for any given boost level. If we want to draw a line into the compressor map for our engines needs we need to calculate the needed airflow for several boost levels.

This all results in 2 simple lines in the compressor map which indicate the maximum flow required from the turbo by our engine. Now we can clearly see where out engine leaves the compressor map and thus the limit for the combination of the two (engine and turbo) lies. We also see that – even for the 2.3 litre engine – the TD04 can sustain a much higher boost pressure at higher rpms than the T25 can. Even at 1.4 bar boost (pressure ratio = 2.4) the TD04 is within its limits and would flow approximately 420 cfm. Would we have done the same with the T25 turbo we would most certainly be in the choke area and the turbo would be unable to get us the airflow that we required.

For selecting the best wheel trim-housing, see [**isaac-lowry_turbo_2004**].

Boost (bar)	Boost (psi)	2.0L lb/min	2.0L cfm	2.3L lb/min	2.3L cfm
0.2	2.9	3.7	53.1	4.2	61
0.4	5.8	7.3	106.1	8.4	122
0.6	8.7	11	159.2	12.6	183
0.8	11.6	14.7	212.2	16.9	244
1.0	14.5	18.3	265.3	21.1	305
1.2	17.4	22	318.3	25.3	366
1.4	20.3	25.7	371.4	29.5	427
1.6	23.2	29.3	424.4	33.8	488

Table B.2: Airflow in cfm and lb/min as given by equation (B.6)

Figure B.2: turbo-lines

B.3 Conclusion

Comparing the two compressor maps for TD04 and T25 (see appendix C) we can clearly see that the TD04 can flow more air at a higher pressure ratio and with a higher efficiency. Given the fact that the turbine blades are larger than in the T25, spool up will be a bit slower, but high end power will be much better. Upgrading your turbo will affect more than meets the eye. The VE map in the Trionic would probably need adjustments because the hardware in the airflow has been changed. This means the volumetric efficiency also changes and thus the correction table needs changing too. Also, when boost values rise, the intercoolers capacity for air flow comes into play. You must make sure that the intercooler is not so restrictive that upgrading the turbo will result in a burst intercooler. A high capacity cross flow intercooler would be a good option here. And last but not least, upgrading the turbo charger means – that is the goal here – more air flow to the cylinders and thus more oxygen to burn. If we upgrade the turbo we need to consider the injectors too. If the injectors can't flow the amount of fuel needed to burn the amount of oxygen pushed into the cylinders we would have gained nothing.

Appendix C

Specifications of common Saab turbochargers

Needs the GT

Import from p

Appendix D

More information on Saab engine design

A previous version of this document contained information that was copied from other published sources without permission. For information on knock and misfire detection, see **eriksson_closed_1998** [**eriksson_closed_1998**]. The section on intercoolers can be found in **ferozepuria_turbocharger** [**ferozepuria_turbocharger**].

Index