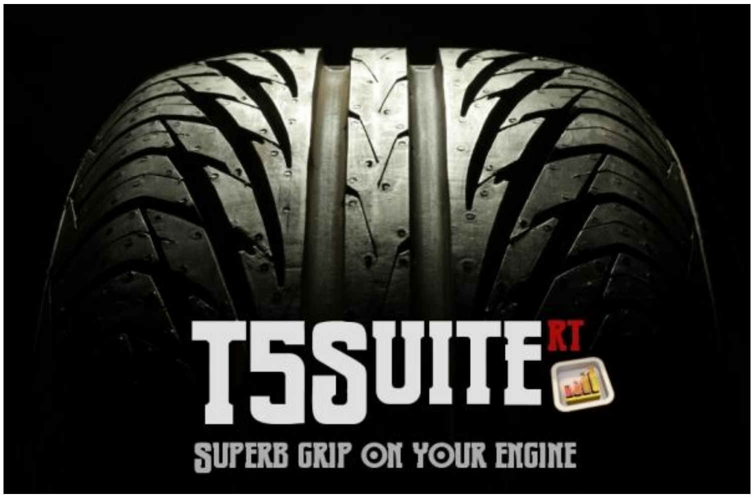
Trionic 5



Preface

This document is intended for Saab fanatics and engineers who want to start understanding the Saab Trionic 5 engine 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 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 T5Suite software will be referenced. This software will enable you to really "get into" the Trionic. The T5Suite software can be downloaded from the T5Suite website.

http://trionic.mobixs.eu

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, Renton, Sourcode, Seb, Sandy Rus, JKB, L4staero and Steve Hayes.

These icons are used throughout the document to denote:

References

T5.2 remarks

Advanced technical topics

Table of contents

Hardware

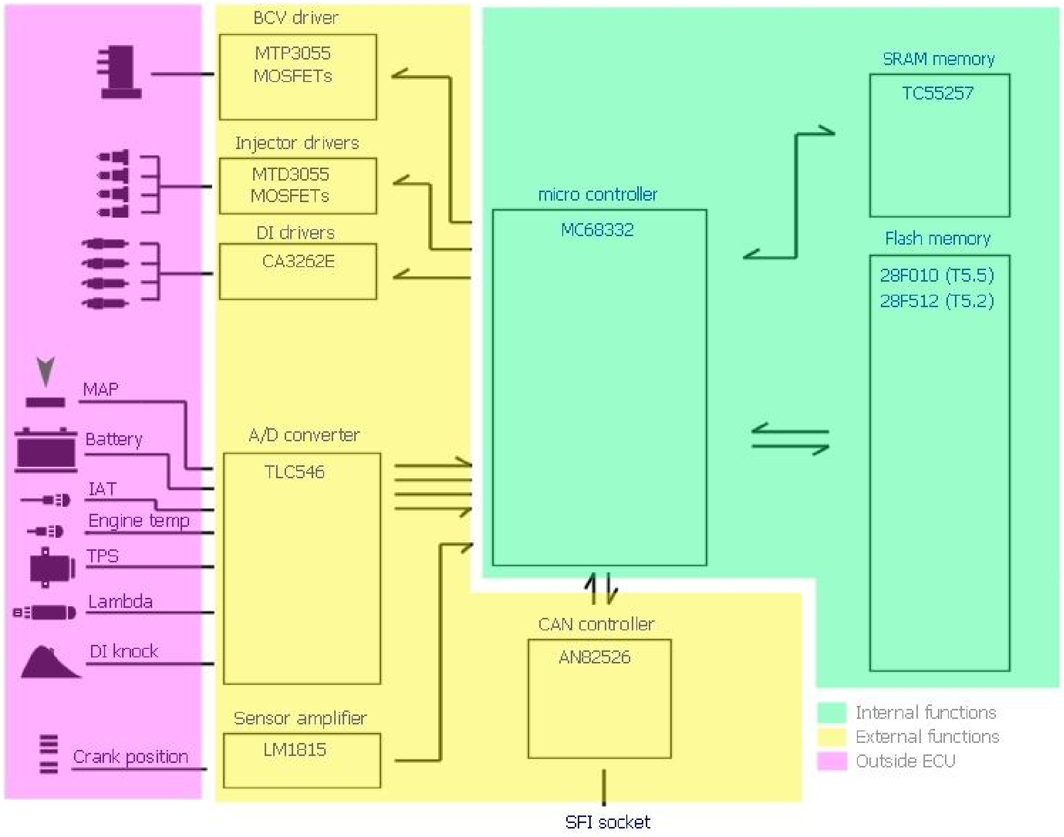
The T5 is build around a Motorola MC68332 microcontroller. This is a 32 bit controller that handles the entire engine management including fuel injection, ignition timing, knock control, diagnostics and boost pressure control.
The processor has a large 2Mb (256 Kbyte) flash memory (only 1Mb for T5.2) to its disposition for fetching program code and maps. This memory is divided into two physical flash chips (28F010 for T5.5, 28F512 for T5.2) that each holds half of the program memory. Saab has made it a little harder to reverse engineer the T5 by having all even bytes of program memory written to one chip and all uneven bytes to the other. So, byte 0 is in the first chip whereas byte 1 is in the second one, and so on. Of course some RAM memory is present in the form of a 256 Kbit SRAM TC55257 chip. This results in 32 Kbyte of RAM. The board also contains a AN82526 (CAN Controller), a TLC546 (8 bit A/D converter), and a LM1815 (Sensor amplifier). This sensor amplifier is used for crank shaft position sensing.

Integrated circuit list

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

Partnumber	Function	Usage	# on board
MC68332	Main micro controller	All functions	1
28F010	1 Mb flash memory	Storage of program	2
TC55257	256 kb SRAM	Runtime memory	1
L4937	Dual multifunction voltage regulator	Power supply on board	1
AN82526	CAN interface	Communication	1
TLC546	A/D converter (19 inputs)	Sensor reading	1
LM1815	Sensor amplifier	Crankshaft sensor	1
L9842	Octal Power Driver		2
74HC00	Quad positive NAND gate		
74HC02	Quad NOR gate		
74HC03	Quad NAND gate (open drain)		
74HC14	6 Hex inverting schmitt triggers		
74HC74	Dual D type flip-flop		
74HC123	Dual monostable multivibrator		
CA3262E	Quad 700 mA driver	DI triggering	1
TLC274	Quad op-amp		
MTP3055V	N-channel MOSFET TO-220	BCV drivers (high current)	3
MTD3055VL	N-channel MOSFET DPAK SMD	4 Injector drivers 2 EVAP purge valve drivers 1 O2 sensor preheating 1 Vacuum pump driver 1 IAC valve driver	10

Block schematic diagram



In 1994- Trionic (T5.5) there is a slotted disk attached to crankshaft which is read with an inductive pickup sensor. Each time a slot and a bridge in the disk moves past the sensor the reluctance of the magnetic circuit variates and thus a signal can be acquired. There is one bridge missing in this disk and that is used to determine the TDC. The signal from the sensor is converted to a usable signal for the ECU by the LM1815.

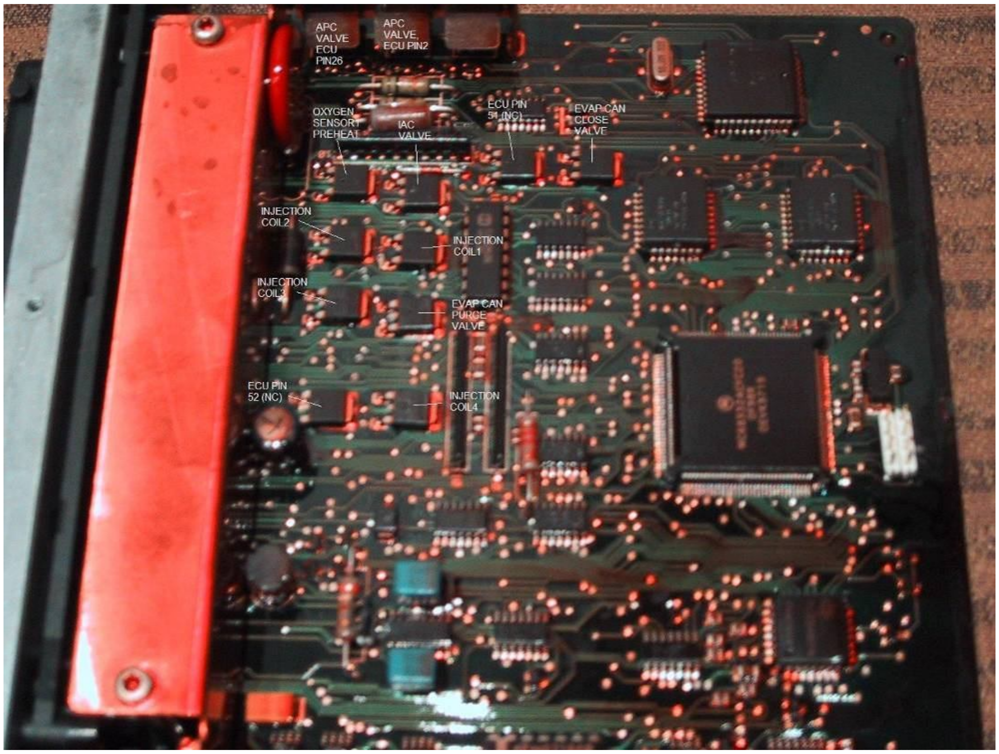
Trionic is an all sequential injection system meaning that each dose of fuel is calculated for each cylinder in ignition sequence. There is a lot of sensor data (like engine coolant temperature, manifold absolute pressure (MAP), manifold air temperature (MAT)) that is used for calculating the injector open time. Also, several constants are used like engine volumetric efficiency at a certain point of load (a matrix of rpm and manifold pressure), injector constant, battery voltage table etc. Even further, this is still adjusted by variables like lambda feedback and throttle position sensor delta from previous position of it (acceleration enrichment and deceleration enleanment).

There are three different hardware versions of T5. In 1993 Trionic 5.2 was used with engines B324R and B234L. In 1994 Trionic 5.5 came to all engines (in 9000, NG900 turbo only) and from 1996 on Trionic 5.5 was updated with faster CPU (20 MHz, to cope with ODBII requirements) and flash memories (90 ns access time).

The ignition switch (+15, pin 60) on the Trionic 5 box controls the internal +12 volt regulator. When you turn off the ignition, the CPU and others will power down too. There is apparently a delay circuitry on the board which will turn on the box after about 15 minutes for updating adaptive injection tables etc.

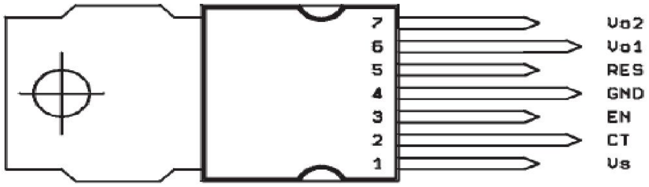
PCB details

The PCB layout is not entirely known of course because SAAB did not release details about this, even in the service manuals. Finding out how things are setup is not so very difficult though, once you know what the system should do and what hardware components are on the board.
The image will give you some idea on what is what on the board.



Power supply

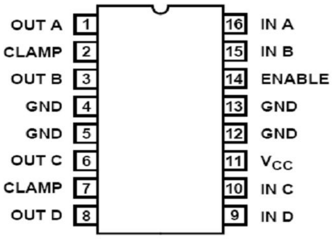
The board is powered by the car battery. The input voltage (+11V - +14V) is transformed to a usable voltage by the L4937 (Voltage regulator). The input voltage (from the battery) is supplied to pinnumber 1 on the regulator. The output voltages (Vo1 and Vo2) are stabilized +5 volt which can sink 50 mA and 500 mA respectively. The Vo1 is a standby voltage. If the ignition is turned off an RC network will keep the power supply in operational state for a while and then lets it resume into standby modus in which the power drain is significantly lower. The maximum standby current the regulator can sink is 50 mA. Nice to know is that the regulator can supply its +5 volt output when the input ranges from +6V up to +28V (with transient spikes up to +40V). When starting a car the battery voltage can drop as low as 6-7 volts when the battery is not in tiptop shape. The regulator is located opposite to the MOSFETs that drive the boost control valve.



DI cartridge triggering

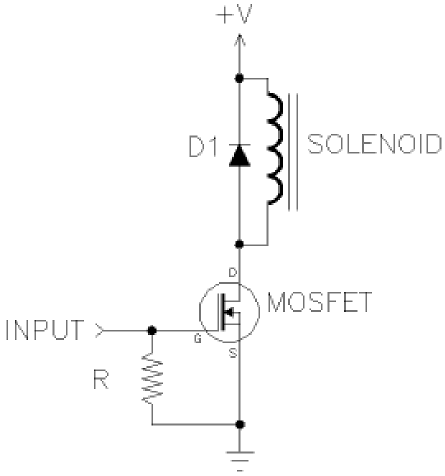
The DI cartridge has a trigger input for firing the four individual sparkplugs. These are triggered by signals from the ECU on pin 9, 10, 11 and 12 which are generated in the power driver IC CA3236 on the Trionic PCB (topside, 16 pin DIL housing). Internally these four pins are connected as shown in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

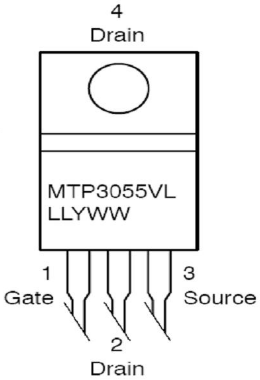
DI cartridge pin	ECU pinnumber	CA3236 pinnumber	Description
2	9	1 (OUT A)	Trigger cylinder 1
3	10	3 (OUT B)	Trigger cylinder 2
4	11	6 (OUT C)	Trigger cylinder 3
5	12	8 (OUT D)	Trigger cylinder 4



BPC drivers (MTP3055V)

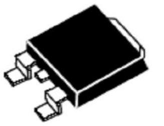
The ECU pulses the boost control valve in order to bleed air pressure to the wastegate or to the compressor housing and thus controlling the charging pressure of the turbo. The boost control valve is simply a set of coils that control the position of a metal plate in the valve to determine how much air should pass in one direction and how much air should pass in the other direction. Driving the coils with enough energy for them to be able to shift the plate to one side or the other needs more current than the microcontroller can deliver by itself. The outputs of the controller could never sink enough current for the coils so a current amplifier is needed. The FETs (Field Effect Transistors) used by default don't conduct current. When a tiny amount of current starts flowing through the gate suddenly the circuit Source – Drain will conduct current. In this way we have a current amplifier (also called electronically activated switch, as one could indeed characterize a FET) and the microcontroller can determine when to flow large amount of current (and thus energy) though the coils in the valve. Pulsing the coils in the right frequency and pulse width now controls the amount of air pressure applied to the wastegate actuator.
The image below shows the FET used for driving the coils in the valve and a schematic diagram of how this is setup to work.





Injector drivers (MTD3055VL)

Since fuel injection also relies on a solenoid valve to open and close the injector Trionic needs power drivers for these too. These smaller packages can supply the same amount of current as the TO-220 packaged ones that are used for the boost control valve, but the resistance in the conducting state is slightly higher. The way the MOSFETs are used is very similar to the boost control valve.

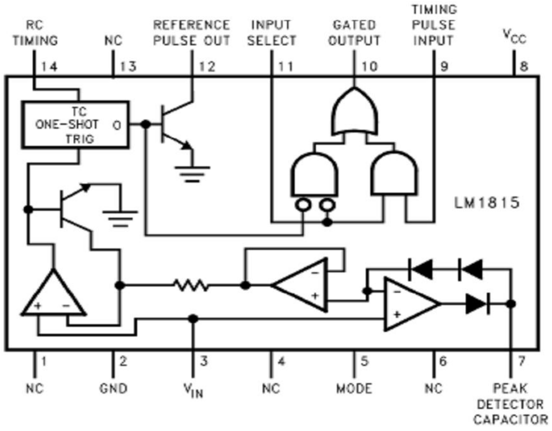


STYLE 2:

PIN 1.	GATE
2.	DRAIN
3.	SOURCE
4.	DRAIN

Crankshaft position sensor (LM1815)

The crankshaft sensor signal needs to be amplified before the microcontroller can interpret it. This amplification is done by the LM1815. It is probably used in adaptive mode (mode 1). In this setup the signal from the VR pickup sensor for the crankshaft position is (filtered) supplied to pin 3 (Vin). The nice square wave signal is picked up from pin 12 (reference pulse out) and supplied to the microcontroller for timing and counting.



Analogue to Digital converter (TLC546)

input#	HW address	ECU pin	TLC546 pin	Description
0			1 (INPUTA0)	AD_trot, Throttle Position, 0-5 volt, 0-255
1			2 (INPUTA1)	
2	0xFFFD0D	68	3 (INPUTA2)	AD_kylv, Coolant temp, 0-5 volt, 0-255
3			4 (INPUTA3)	
4	0xFFFD0B	44	5 (INPUTA4)	AD_knock
5	0xFFFD08	46	6 (INPUTA5)	AD_lufttemp
6			7 (INPUTA6)	
7			8 (INPUTA9)	
8	0xFFFD05	23	9 (INPUTA8)	AD_sond (0-1.3 volt, 0-80)
9	0xFFFD07	70	10 (INPUTA9)	AD_cat (0-1.3 volt, 0-69)
10			11 (INPUTA10)	
11			12 (INPUTA11)	
12			13 (INPUTA12)	
13			15 (INPUTA13)	
14			16 (INPUTA14)	
15	0xFFFD03	69	17 (INPUTA15)	AD_EGR (0-5 volt, 0-255)
16			18 (INPUTA16)	
17			19 (INPUTA17)	
18			20 (INPUTA18)	

Flash

Important note!
The flash memories in the T5 ECUs are starting to get old and writing to the flash does not always work. If this should be the case the ECU will be inoperable and the only way to fix it is to physically replace the flash chips on the board. Unfortunately there is no way to determine the status of the flash in advance. Please do not flash if you only have one ECU and you really need it to drive your car.

The binary file that consists of 256 kilobyte of program code (128 kB for T5.2) can be downloaded from the ECU in several ways. One option is to remove the PLCC32 flash chips from the board and read the program code from the chips in a programmer (for example the well known Willem programmer). Because this requires special tools (or a very skillful hand) that are not very common this is not the ideal way. The advantage is that sockets can be mounted on the board once the chips have been desoldered. This way, new firmware can be flashed with a normal flash programmer.
The other way to download the code is to hook up a BDM (background debugger module) to the ECU and use BD32.exe to download the code from the flash with the ECU running (not in a car though, on the bench ☺). Last but not least, the binary file can be downloaded from the ECU through a Canbus interface for which you need an adapter and some wiring. More will be written about this later in this document.

Using public domain LPT BDM interface

To download the code you'll need to have a T5 on your desk, a power supply that can supply +12 volts regulated, some wiring, **a Win9x (!!!) or DOS PC** with a parallel port, a BDM interface (see appendix III for more information), a 8 pin DIL header, a soldering iron, the BDM software (http://trionic.mobixs.eu/bd32-122.zip), the T5dumper scripts available from the http://trionic.mobixs.eu and some patience.

- Open the ECU by removing the six small torx screws.
- Locate the 8 pin header pads on the top side of the board and solder the 8 pin header onto it.
- Connect the power wires to the ECU as shown in *image 1*. See appendix II for more information about T5 pinout.
- Connect the BDM interface to your computers parallel port.
- Connect the BDM interface to the 8 pin header you've soldered onto the board. See appendix III for information about pinout.
- Power up the ECU.

Prerequisites

- BD32 must be installed and working (see appendix XII).
- The BDM interface must be connected to the T5 ECU and to the computer.
- The T5 ECU must be powered up.

Install FlashT5

- Download FlashT5 from http://trionic.mobixs.eu
- Unpack the files in the same directory as BD32 is installed in.

Prepare ECU for flash management

- In BD32 give the command: "do prepT5.do". This will start a script that prepares the processor in the ECU for flash management. Among several things this script will disable the watchdog, set up registers to allow writing to flash and tell BD32 where to load the target resident program.

Dumping flash memory

This step is not necessary but can be used to verify that the content of the flash really is changed. It is a good idea to always make a backup of the flash before you start writing to it.

- In BD32 give the command: "dumpT5 filename.bin".

filename.bin can be replaced with a file name of your choice as long as you stick with the 8.3 format (so 8 characters for filename and 3 for extension). BD32 will now download a target resident program that reads from flash and sends the information to BD32. You will see a number of dots appearing in BD32 during the copying. When you return to the BD32 prompt you will have successfully downloaded the flash content.

Flashing the ECU

- Read Important note above!
- In BD32 give the command: "flashT5 filename.bin"

filename.bin should be replaced by the name of the binary file that you want to write to flash (stick with 8.3 format here as well). BD32 will now download a target resident program flashT5.d32 to the processor in the ECU and execute it. The target resident program and BD32 cooperates during the flashing (BD32 takes care of file handling and the target resident program handles writing to flash). During the flashing you will see a number of dots being printed in BD32. This will stop when the flashing is completed and the command prompt of BD32 will appear.

If you run into a problem

If you run into problem when you are flashing try this:

- Double the delay parameter.
- The bottom field in BD32 should report "MCU: STOPPED" while you are using flashT5. If you for some reason have given the reset command you need to run the prepT5.do script again in order to setup the registers and stop the MCU.
- If you get an Erase Error or Program Error message it is likely that your flash cannot be written to (see Important note above). You can try flashing a couple of more times but it's likely that the ECU no longer can be used.

5⚠2

For T5.2 there is a different script (dumpt52) which is available from http://trionic.mobixs.eu/FLASHT5ANDT52.zip

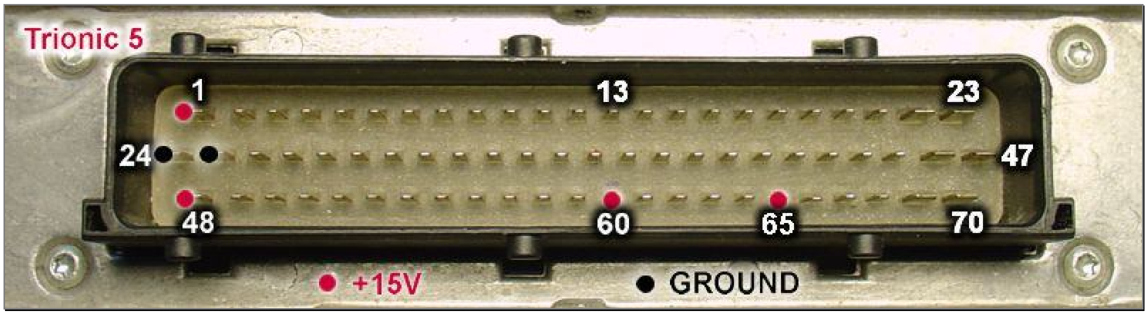
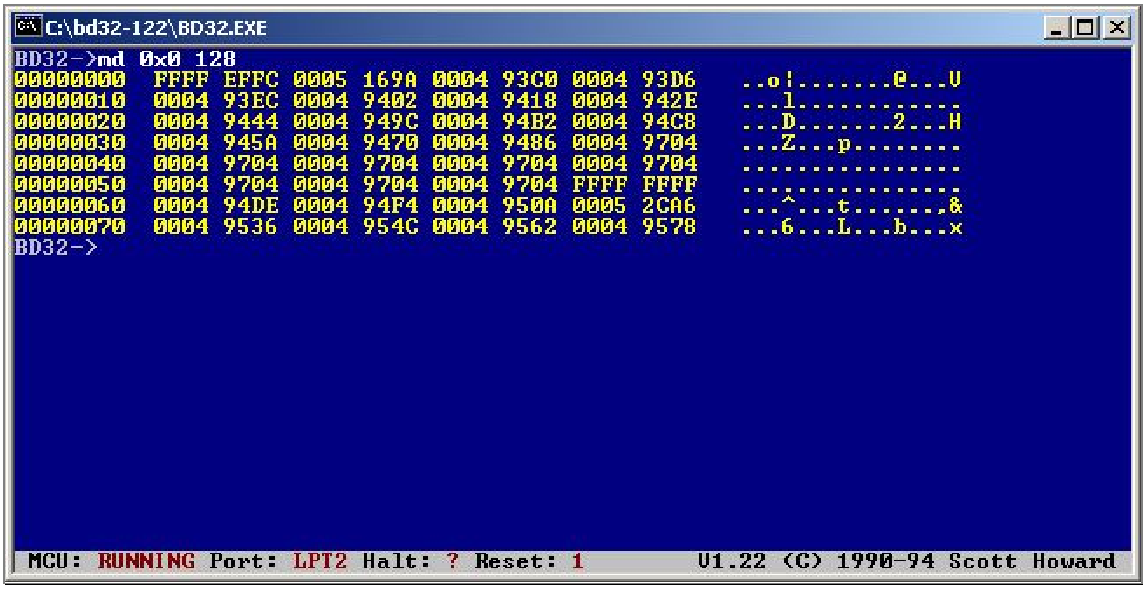


Image 1: T5.5 pinout



```
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6   ..o!.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E   ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8   ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704   ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704   ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF   ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6   ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578   ...6...L...b...x
BD32->
```

```
MCU: RUNNING Port: LPT2 Halt: ? Reset: 1        V1.22 (C) 1990-94 Scott Howard
```

Image 2: BD32 memory display result

Downloading with PEMicro USB BDM interface

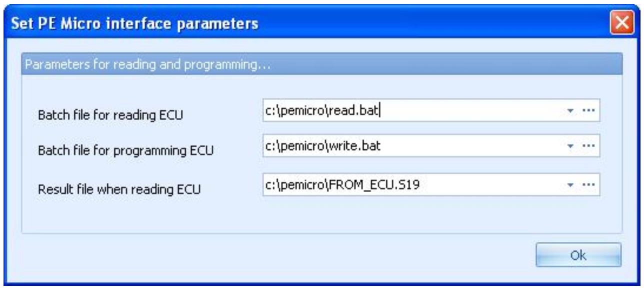
Install the PEMicro device driver as described in the PEMicro documentation.
Install the PEMicro software as described in the PEMicro documentation.
Create a read batchfile for your settings as shown below in an example.
The read batchfile should be something like this.

```
c:\pemicro\pkg32z\cprog32z.exe c:\pemicro\READ_AMD.CFG Interface=USBMULTILINK Port=USB1 bdm_speed 2 ?
```

The READ_AMG.CFG file should contain these statements:

```
RE
CM c:\pemicro\T5_AMDalgo.32P 0
UM c:\pemicro\FROM_ECU.S19
```

The read batchfile should be tailored to your specific needs. If you installed the PEMicro software in another location then C:\Pemicro, the batchfile should be altered likewise.
Once you've got the batchfile running you can enter it as a parameter in T5Suite.
The config file tells which filename should be used by the PEMicro software. In the example the filename is FROM_ECU.S19. Make sure this result file is the same as in the read batchfile.


Set PE Micro interface parameters

Parameters for reading and programming...

Batch file for reading ECU: c:\pemicro\read.bat
Batch file for programming ECU: c:\pemicro\write.bat
Result file when reading ECU: c:\pemicro\FROM_ECU.S19

Flashing with PEMicro USB BDM interface

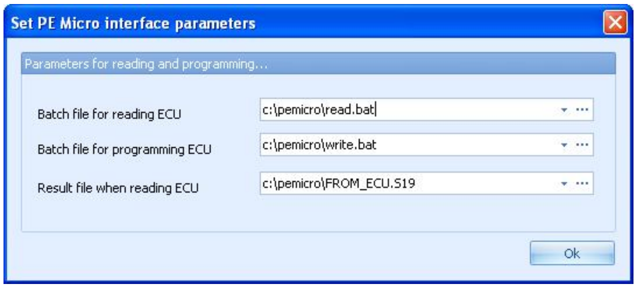
Create a write batchfile for your settings as shown below in an example.
The write batchfile should be something like:

```
c:\pemicro\pkg32z\cprog32z.exe c:\pemicro\WRITE_AMD.CFG Interface=USBMULTILINK Port=USB1 bdm_speed 2 ?
```

And the WRITE_AMD.CFG file:

```
RE
CM C:\pemicro\T5_AMDalgo.32P 0
EM
BM
SS c:\pemicro\TO_ECU.S19
PM
VM
RE
```

T5Suite creates a TO_ECU.S19 automatically when writing to the ECU so this filename should also be used in the write config file.


Set PE Micro interface parameters

Parameters for reading and programming...

Batch file for reading ECU: c:\pemicro\read.bat
Batch file for programming ECU: c:\pemicro\write.bat
Result file when reading ECU: c:\pemicro\FROM_ECU.S19

Checksum

The firmware uses a checksum over the flash contents.
This checksum is a simple 32 bit addition of all bytes in the flash up to the part where the program code ends. The end of the program code can be found programmatically by reading a footer field with identifier 0xFE. This field indicates the last byte in the binary that belongs to the program code. This byte is the last byte that should be calculated in the checksum.
Normally this is immediately after a certain marker. This marker is 4E FA FB CC for normal binaries. If you alter any code between 0x00000 and the endmarker the checksum will have to be updated. The checksum verification and update routines are also implemented in T5Suite.

Firmware

General

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 IDAPro. 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 symboltable inside the binary.

Memory map

T5.5
0x00000 – 0x07FFF = SRAM
0x40000 – 0x7FFFF = FLASH (program memory)

⚠ *T5.2*

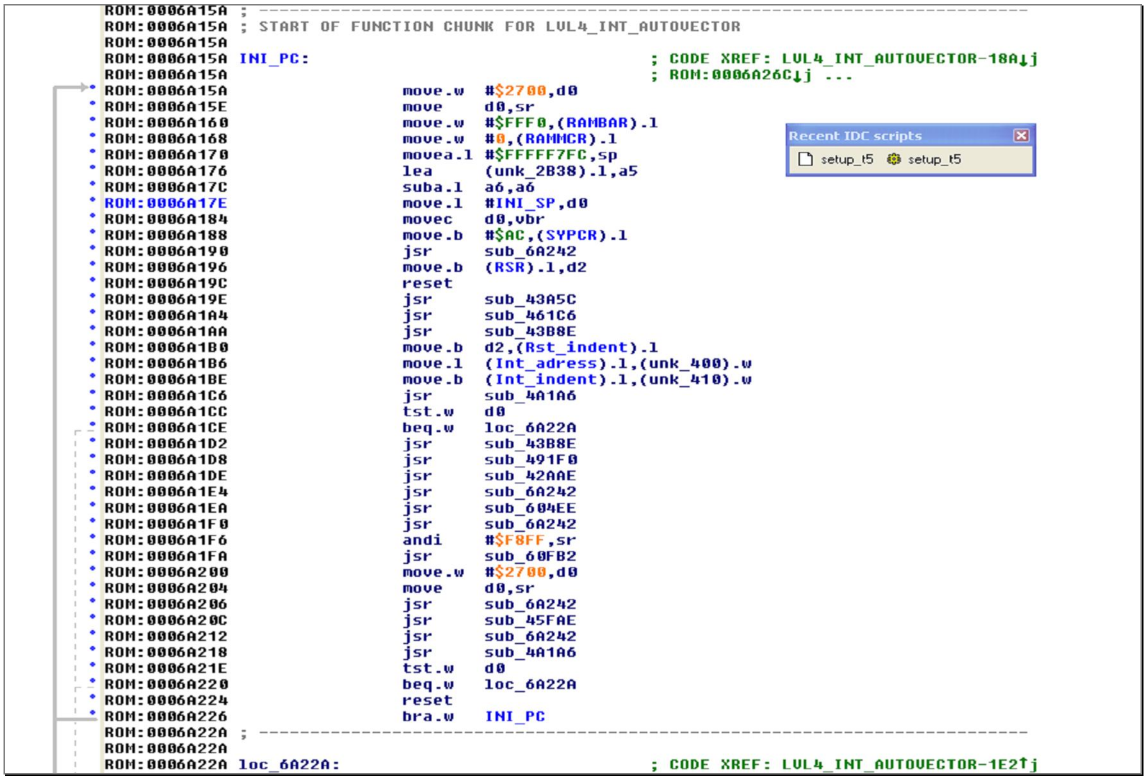
0x00000 – 0x07FFF = SRAM
0x60000 – 0x7FFFF = FLASH (program memory)

Disassembling the code

To convert the binary flash file to readable assembler language we now have to disassemble the binary. This can be done by any 68332 disassembler but I recommend using IDA Pro (Interactive Disassembler Professional).
To get the code disassembled in IDA Pro you need to do at least these steps.

- Open the bin file in IDA Pro.
- Select processor type Motorola 68300 series (this tells the program which disassembler to use).
- Set up memory maps for RAM and ROM (offset and length), so RAM = 0x00000 to 0x40000 and ROM 0x40000 to 0x80000. (⚠ for T5.2: 0 – 0x60000 and 0x60000 to 0x80000).
- Look up the second double word (after FFFF F7FC) in hex view in the binary file, this is the boot vector for the binary.
- Go to the address given by the second double word in IDA Pro.
- Press C (starts analysis).

There is an easier way to do this but you will need a script that has been made by sandy_rus which is available from http://www.ecuproject.com.
Another route is to use the – simpler – disassembler that has been build into T5Suite.



```
ROM:0006A15A ; ---------------------------------------------------------------------------
ROM:0006A15A ; START OF FUNCTION CHUNK FOR LVL4_INT_AUTOVECTOR
ROM:0006A15A
ROM:0006A15A INI_PC:                                 ; CODE XREF: LVL4_INT_AUTOVECTOR-18A↓j
ROM:0006A15A                                         ; ROM:0006A26C↓j ...
ROM:0006A15A                 move.w  #$2700,d0
ROM:0006A15E                 move    d0,sr
ROM:0006A160                 move.w  #$FFF0,(RAMBAR).l
ROM:0006A168                 move.w  #0,(RAMMCR).l
ROM:0006A170                 movea.l #$FFFFF7FC,sp
ROM:0006A176                 lea     (unk_2B38).l,a5
ROM:0006A17C                 suba.l  a6,a6
ROM:0006A17E                 move.l  #INI_SP,d0
ROM:0006A184                 movec   d0,vbr
ROM:0006A188                 move.b  #$AC,(SYPCR).l
ROM:0006A190                 jsr     sub_6A242
ROM:0006A196                 move.b  (RSR).l,d2
ROM:0006A19C                 reset
ROM:0006A19E                 jsr     sub_43A5C
ROM:0006A1A4                 jsr     sub_461C6
ROM:0006A1AA                 jsr     sub_43B8E
ROM:0006A1B0                 move.b  d2,(Rst_indent).l
ROM:0006A1B6                 move.l  (Int_adress).l,(unk_400).w
ROM:0006A1BE                 move.b  (Int_indent).l,(unk_410).w
ROM:0006A1C6                 jsr     sub_4A1A6
ROM:0006A1CC                 tst.w   d0
ROM:0006A1CE                 beq.w   loc_6A22A
ROM:0006A1D2                 jsr     sub_43B8E
ROM:0006A1D8                 jsr     sub_491F0
ROM:0006A1DE                 jsr     sub_42AAE
ROM:0006A1E4                 jsr     sub_6A242
ROM:0006A1EA                 jsr     sub_604EE
ROM:0006A1F0                 jsr     sub_6A242
ROM:0006A1F6                 andi    #$F8FF,sr
ROM:0006A1FA                 jsr     sub_60FB2
ROM:0006A200                 move.w  #$2700,d0
ROM:0006A204                 move    d0,sr
ROM:0006A206                 jsr     sub_6A242
ROM:0006A20C                 jsr     sub_45FAE
ROM:0006A212                 jsr     sub_6A242
ROM:0006A218                 jsr     sub_4A1A6
ROM:0006A21E                 tst.w   d0
ROM:0006A220                 beq.w   loc_6A22A
ROM:0006A224                 reset
ROM:0006A226                 bra.w   INI_PC
ROM:0006A22A ; ---------------------------------------------------------------------------
ROM:0006A22A
ROM:0006A22A loc_6A22A:                              ; CODE XREF: LVL4_INT_AUTOVECTOR-1E2↑j
```

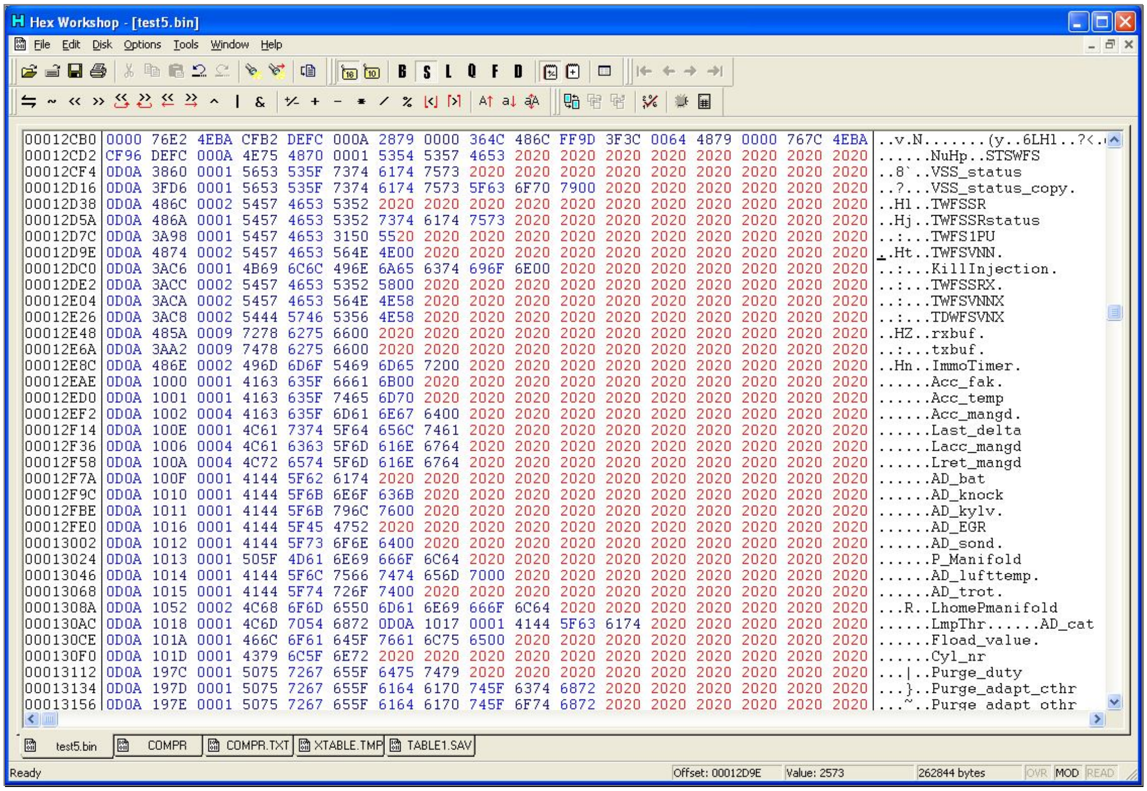
Image 3: Result of disassembly

Symbol tables

General

Each T5 firmware file contains a symbol table describing data structures in the program. These symbols can be extracted together with their corresponding memory addresses (ROM and RAM).
The SIM - "System Integration Module" - registers in the MC68332 show how the memory is mapped. See Motorola user manual for register addresses. For T5.5 the RAM starts at 0x00000 and the ROM starts at 0x40000. 5.2 T5.2 ROM start address is 0x60000.
Image 4 gives a general idea on what these symbol tables look like.

Column 1: Symbol delimiter (0D0A)
Column 2: Symbol SRAM address
Column 3: Symbol Map number of bytes



```
00012CB0 0000 76E2 4EBA CFB2 DEFC 000A 2879 0000 364C 486C FF9D 3F3C 0064 4879 0000 767C 4EBA ..v.N.......(y..6LHl..?<..
00012CD2 CF96 DEFC 000A 4E75 4870 0001 5354 5357 4653 2020 2020 2020 2020 2020 2020 2020 2020 ......NuHp..STSWFS
00012CF4 0D0A 3860 0001 5653 535F 7374 6174 7573 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..8`..VSS_status
00012D16 0D0A 3FD6 0001 5653 535F 7374 6174 7573 5F63 6F70 7900 2020 2020 2020 2020 2020 2020 ..?...VSS_status_copy.
00012D38 0D0A 486C 0002 5457 4653 5352 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..Hl..TWFSSR
00012D5A 0D0A 486A 0001 5457 4653 5352 7374 6174 7573 2020 2020 2020 2020 2020 2020 2020 2020 ..Hj..TWFSSRstatus
00012D7C 0D0A 3A98 0001 5457 4653 3150 5520 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..:...TWFS1PU
00012D9E 0D0A 4874 0002 5457 4653 564E 4E00 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..Ht..TWFSVNN.
00012DC0 0D0A 3AC6 0001 4B69 6C6C 496E 6A65 6374 696F 6E00 2020 2020 2020 2020 2020 2020 2020 ..:...KillInjection.
00012DE2 0D0A 3ACC 0002 5457 4653 5352 5800 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..:...TWFSSRX.
00012E04 0D0A 3ACA 0002 5457 4653 564E 4E58 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..:...TWFSVNNX
00012E26 0D0A 3AC8 0002 5444 5746 5356 4E58 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..:...TDWFSVNX
00012E48 0D0A 485A 0009 7278 6275 6600 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..HZ..rxbuf.
00012E6A 0D0A 3AA2 0009 7478 6275 6600 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..:...txbuf.
00012E8C 0D0A 486E 0002 496D 6D6F 5469 6D65 7200 2020 2020 2020 2020 2020 2020 2020 2020 2020 ..Hn..ImmoTimer.
00012EAE 0D0A 1000 0001 4163 635F 6661 6B00 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Acc_fak.
00012ED0 0D0A 1001 0001 4163 635F 7465 6D70 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Acc_temp
00012EF2 0D0A 1002 0004 4163 635F 6D61 6E67 6400 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Acc_mangd.
00012F14 0D0A 100E 0001 4C61 7374 5F64 656C 7461 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Last_delta
00012F36 0D0A 1006 0004 4C61 6363 5F6D 616E 6764 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Lacc_mangd
00012F58 0D0A 100A 0004 4C72 6574 5F6D 616E 6764 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Lret_mangd
00012F7A 0D0A 100F 0001 4144 5F62 6174 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_bat
00012F9C 0D0A 1010 0001 4144 5F6B 6E6F 636B 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_knock
00012FBE 0D0A 1011 0001 4144 5F6B 796C 7600 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_kylv.
00012FE0 0D0A 1016 0001 4144 5F45 4752 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_EGR
00013002 0D0A 1012 0001 4144 5F73 6F6E 6400 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_sond.
00013024 0D0A 1013 0001 505F 4D61 6E69 666F 6C64 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......P_Manifold
00013046 0D0A 1014 0001 4144 5F6C 7566 7474 656D 7000 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_lufttemp.
00013068 0D0A 1015 0001 4144 5F74 726F 7400 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......AD_trot.
0001308A 0D0A 1052 0002 4C68 6F6D 6550 6D61 6E69 666F 6C64 2020 2020 2020 2020 2020 2020 2020 ...R..LhomePmanifold
000130AC 0D0A 1018 0001 4C6D 7054 6872 0D0A 1017 0001 4144 5F63 6174 2020 2020 2020 2020 2020 ......LmpThr......AD_cat
000130CE 0D0A 101A 0001 466C 6F61 645F 7661 6C75 6500 2020 2020 2020 2020 2020 2020 2020 2020 ......Fload_value.
000130F0 0D0A 101D 0001 4379 6C5F 6E72 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ......Cyl_nr
00013112 0D0A 197C 0001 5075 7267 655F 6475 7479 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...|..Purge_duty
00013134 0D0A 197D 0001 5075 7267 655F 6164 6170 745F 6374 6872 2020 2020 2020 2020 2020 2020 ...}..Purge_adapt_cthr
00013156 0D0A 197E 0001 5075 7267 655F 6164 6170 745F 6F74 6872 2020 2020 2020 2020 2020 2020 ...~..Purge_adapt_othr
```

Ready Offset: 00012D9E Value: 2573 262844 bytes

Image 4: Symbol table in T5 firmware

While examining the symbol table you can see that the separator is 0x0d 0x0a.
The next two bytes are the SRAM address where the variable can be found and the two bytes after that are the length of the variable. You can see this clearly where two variables are sequentially in the symbol table. For example:

00ED5E: 19 BC 00 50 41 4D 4F 53 5F 74 65 78 74 00 - ...PAMOS_text.
00ED6E: 1A 0C 00 05 41 4D 4F 53 5F 73 74 61 74 75 73 00 - ....AMOS_status.

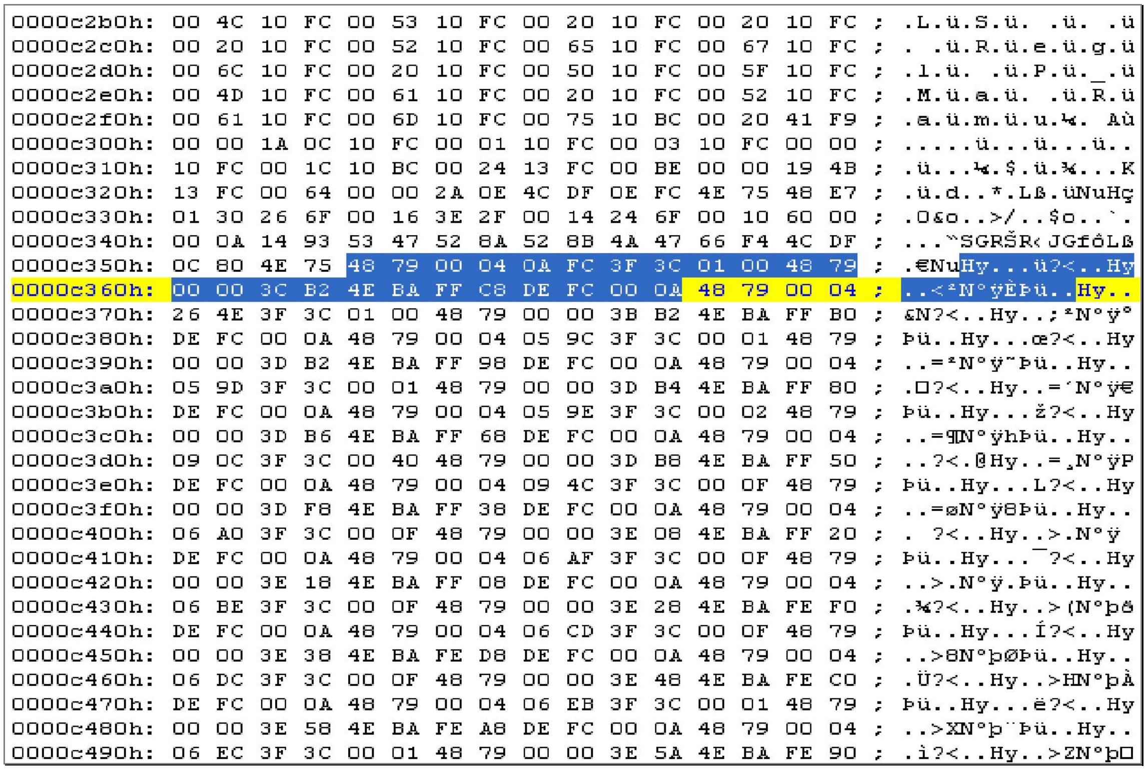
First var: AMOS_TEXT, Starts at 0x19BC with length 0x50.

Adding 0x50 to 0x19BC results in 0x1A0C which is exactly the starting address for the next variable: AMOS_Status which has a length of 0x05.

Note that these are the SRAM addresses in which the runtime instance of the symbol can be found. From a tuning perspective, this is only interesting when we want to alter or view the variables while the car is running. Otherwise, we need the addresses in flash for each symbol.
This can be done by converting the sram address to flash address using the lookuptable that is actually in the flash. The conversion is not really complex once you know how to proceed.
The firmware copies a lot of symbols (tables or maps) to SRAM when the ECU is booted. For the firmware to do this, it needs the addresses for the symbols in SRAM as well as the flash addresses. So, there is a lookup table in the binary that contains all this information.

Finding start of symbol table:
Search binary for string 00 0A 28 79 00. If the byte after this is a 07 the file is T5.2. If the byte after that is 00, the file is T5.5. From the found address find the next 4E 75 record and skip 2 bytes. This is the start of the symboltable.

Finding start of addresslookup table:
Search binary for 4E 75 48 E7 01 30 26 6F 00 16 3E 2F 00 14 24 6F 00 10 60 00 00 0A
From the found address (+ length) find the first : 4E 75 48 79 and we found the start of the addresslookuptable.
In your typical hex editor this will look something like this:



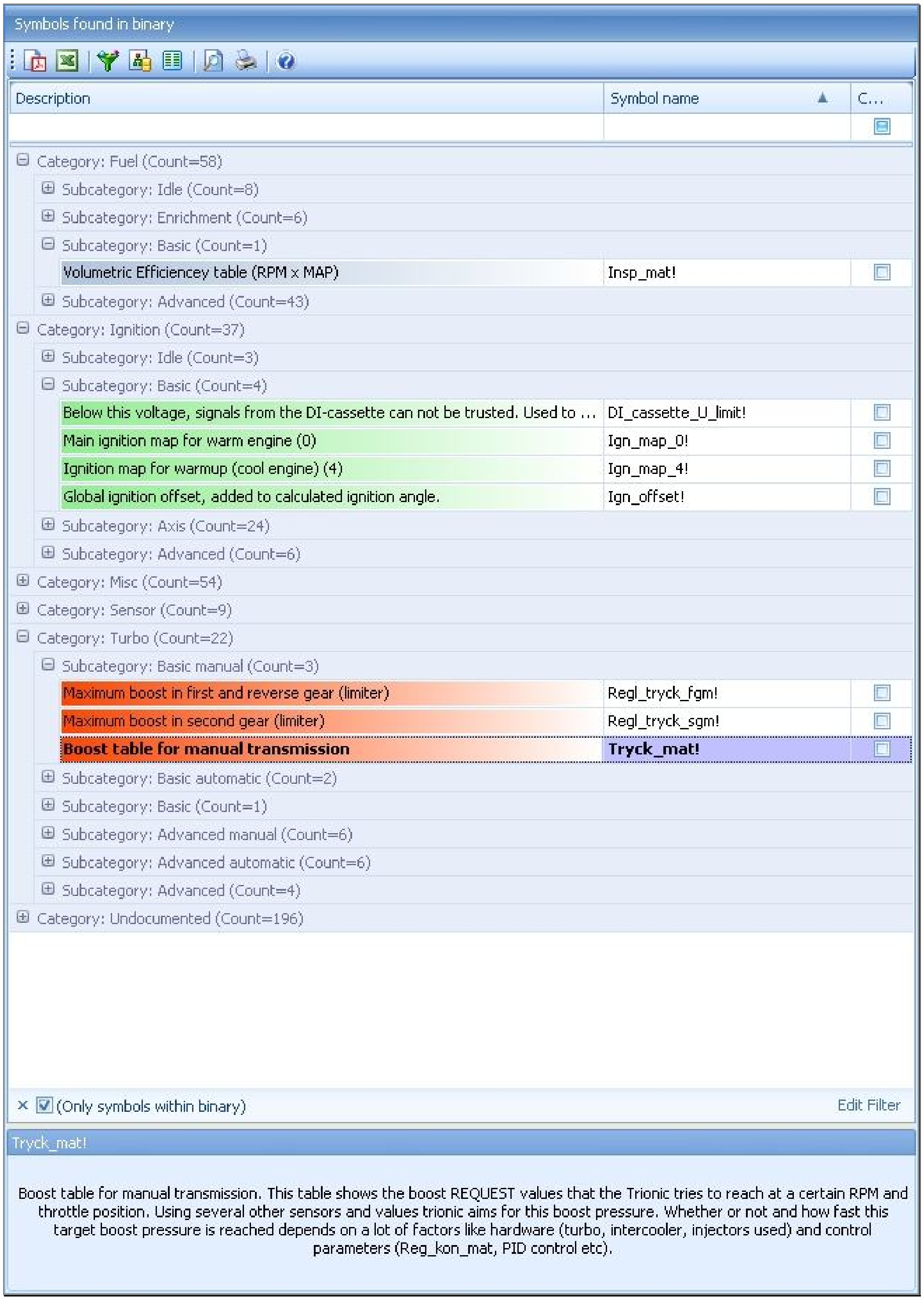
```
0000c2b0h: 00 4C 10 FC 00 53 10 FC 00 20 10 FC 00 20 10 FC ; .L.ü.S.ü. .ü. .ü
0000c2c0h: 00 20 10 FC 00 52 10 FC 00 65 10 FC 00 67 10 FC ; . .ü.R.ü.e.ü.g.ü
0000c2d0h: 00 6C 10 FC 00 20 10 FC 00 50 10 FC 00 5F 10 FC ; .l.ü. .ü.P.ü._.ü
0000c2e0h: 00 4D 10 FC 00 61 10 FC 00 20 10 FC 00 52 10 FC ; .M.ü.a.ü. .ü.R.ü
0000c2f0h: 00 61 10 FC 00 6D 10 FC 00 75 10 BC 00 20 41 F9 ; .a.ü.m.ü.u.¼. Aù
0000c300h: 00 00 1A 0C 10 FC 00 01 10 FC 00 03 10 FC 00 00 ; .....ü...ü...ü..
0000c310h: 10 FC 00 1C 10 BC 00 24 13 FC 00 BE 00 00 19 4B ; .ü...¼.$.ü.¾...K
0000c320h: 13 FC 00 64 00 00 2A 0E 4C DF 0E FC 4E 75 48 E7 ; .ü.d..*.Lß.üNuHç
0000c330h: 01 30 26 6F 00 16 3E 2F 00 14 24 6F 00 10 60 00 ; .0&o..>/..$o..`.
0000c340h: 00 0A 14 93 53 47 52 8A 52 8B 4A 47 66 F4 4C DF ; ...“SGRŠR‹JGfôLß
0000c350h: 0C 80 4E 75 48 79 00 04 0A FC 3F 3C 01 00 48 79 ; .€NuHy...ü?<..Hy
0000c360h: 00 00 3C B2 4E BA FF C8 DE FC 00 0A 48 79 00 04 ; ..<²N°ÿÈÞü..Hy..
0000c370h: 26 4E 3F 3C 01 00 48 79 00 00 3B B2 4E BA FF B0 ; &N?<..Hy..;²N°ÿ°
0000c380h: DE FC 00 0A 48 79 00 04 05 9C 3F 3C 00 01 48 79 ; Þü..Hy...œ?<..Hy
0000c390h: 00 00 3D B2 4E BA FF 98 DE FC 00 0A 48 79 00 04 ; ..=²N°ÿ˜Þü..Hy..
0000c3a0h: 05 9D 3F 3C 00 01 48 79 00 00 3D B4 4E BA FF 80 ; .□?<..Hy..=´N°ÿ€
0000c3b0h: DE FC 00 0A 48 79 00 04 05 9E 3F 3C 00 02 48 79 ; Þü..Hy...ž?<..Hy
0000c3c0h: 00 00 3D B6 4E BA FF 68 DE FC 00 0A 48 79 00 04 ; ..=¶N°ÿhÞü..Hy..
0000c3d0h: 09 0C 3F 3C 00 40 48 79 00 00 3D B8 4E BA FF 50 ; ..?<.@Hy..=¸N°ÿP
0000c3e0h: DE FC 00 0A 48 79 00 04 09 4C 3F 3C 00 0F 48 79 ; Þü..Hy...L?<..Hy
0000c3f0h: 00 00 3D F8 4E BA FF 38 DE FC 00 0A 48 79 00 04 ; ..=øN°ÿ8Þü..Hy..
0000c400h: 06 A0 3F 3C 00 0F 48 79 00 00 3E 08 4E BA FF 20 ; . ?<..Hy..>.N°ÿ 
0000c410h: DE FC 00 0A 48 79 00 04 06 AF 3F 3C 00 0F 48 79 ; Þü..Hy...¯?<..Hy
0000c420h: 00 00 3E 18 4E BA FF 08 DE FC 00 0A 48 79 00 04 ; ..>.N°ÿ.Þü..Hy..
0000c430h: 06 BE 3F 3C 00 0F 48 79 00 00 3E 28 4E BA FE F0 ; .¾?<..Hy..>(N°þð
0000c440h: DE FC 00 0A 48 79 00 04 06 CD 3F 3C 00 0F 48 79 ; Þü..Hy...Í?<..Hy
0000c450h: 00 00 3E 38 4E BA FE D8 DE FC 00 0A 48 79 00 04 ; ..>8N°þØÞü..Hy..
0000c460h: 06 DC 3F 3C 00 0F 48 79 00 00 3E 48 4E BA FE C0 ; .Ü?<..Hy..>HN°þÀ
0000c470h: DE FC 00 0A 48 79 00 04 06 EB 3F 3C 00 01 48 79 ; Þü..Hy...ë?<..Hy
0000c480h: 00 00 3E 58 4E BA FE A8 DE FC 00 0A 48 79 00 04 ; ..>XN°þ¨Þü..Hy..
0000c490h: 06 EC 3F 3C 00 01 48 79 00 00 3E 5A 4E BA FE 90 ; .ì?<..Hy..>ZN°þ□
```

Image 5: Address conversion lookup table

The starting address for the selected symbol is 00040AFC which is in the flash address range (see chapter: memory map). After subtracting 0x040000 (⚠ T5.2: 0x60000) from it this gives us the address in the binary, 0x0AFC. The symbol which belongs to this address has SRAM address 0x3CB2. While writing this document I haven't figured out what the other information contained in the lookup table means.

To save you the time to lookup all addresses manually the T5Suite application will extract all symbol information in one run. Symbol name, SRAM address, flash address and length will be displayed all together.
Some symbols don't have a corresponding flash address although they do have an sram-address. These are runtime only symbols that are used by Trionic (variables, buffers etc).

This *image (6)* will give you an idea of what the symbol table should look like once it has been extracted. See appendix I for a complete list of known symbols.



Symbols found in binary

Description	Symbol name	C...
		□
⊟ Category: Fuel (Count=58)		
⊞ Subcategory: Idle (Count=8)		
⊞ Subcategory: Enrichment (Count=6)		
⊟ Subcategory: Basic (Count=1)		
Volumetric Efficiencey table (RPM x MAP)	Insp_mat!	□
⊞ Subcategory: Advanced (Count=43)		
⊟ Category: Ignition (Count=37)		
⊞ Subcategory: Idle (Count=3)		
⊟ Subcategory: Basic (Count=4)		
Below this voltage, signals from the DI-cassette can not be trusted. Used to ...	DI_cassette_U_limit!	□
Main ignition map for warm engine (0)	Ign_map_0!	□
Ignition map for warmup (cool engine) (4)	Ign_map_4!	□
Global ignition offset, added to calculated ignition angle.	Ign_offset!	□
⊞ Subcategory: Axis (Count=24)		
⊞ Subcategory: Advanced (Count=6)		
⊞ Category: Misc (Count=54)		
⊞ Category: Sensor (Count=9)		
⊟ Category: Turbo (Count=22)		
⊟ Subcategory: Basic manual (Count=3)		
Maximum boost in first and reverse gear (limiter)	Regl_tryck_fgm!	□
Maximum boost in second gear (limiter)	Regl_tryck_sgm!	□
Boost table for manual transmission	**Tryck_mat!**	□
⊞ Subcategory: Basic automatic (Count=2)		
⊞ Subcategory: Basic (Count=1)		
⊞ Subcategory: Advanced manual (Count=6)		
⊞ Subcategory: Advanced automatic (Count=6)		
⊞ Subcategory: Advanced (Count=4)		
⊞ Category: Undocumented (Count=196)		

× ☑ (Only symbols within binary) Edit Filter

Tryck_mat!

Boost table for manual transmission. This table shows the boost REQUEST values that the Trionic tries to reach at a certain RPM and throttle position. Using several other sensors and values trionic aims for this boost pressure. Whether or not and how fast this target boost pressure is reached depends on a lot of factors like hardware (turbo, intercooler, injectors used) and control parameters (Reg_kon_mat, PID control etc).

Image 6: Screenshot of a part of the symbol table

When the user double clicks one of the symbols that has a flash address attached to it, T5Suite will display the corresponding symbol in a viewer. This viewer will display the data in table form was well as in graphical form.

Maps

General

A lot of maps in the T5 are not only made up of a piece of raw data. It also includes x-axis and y-axis information. For example, the ignition map is called ign_map_0!. There are also 2 symbols called ign_map_0_x_axis and ign_map_0_y_axis. These last two symbols hold the axis configuration for the ignition map.

Adaption limiting

The system continuously adapts tables that it holds in SRAM to fine-tune the injection, boost and ignition to the current hardware setup, the fuel used etc.
It uses some initial values in adaption tables – mostly 0x80 – to which it adds or from which it subtracts steps to get the engine running as smooth and economically as possible. To prevent the values to fall or rise too much, mostly a maximum and minimum is set. This is shown by the symbols "Adapt_injfaktor_low!" and "Adapt_injfaktor_high!" perfectly.
In a sample binary these values are: 0x66 and 0x9A. The initial adaption value is 0x80 in symbol "Adapt_inj_imat!".
Another sample is symbol "Adapt_korr!" which holds an entire (initial) adaption map in flash. It is copied to sram when the system first boots and the table is empty. It is filled with 0x80 in flash. The symbols "Adapt_korr_low!" (0x4D) and "Adapt_korr_high!" (0xB3) hold the minimum and maximum values for this adaption map. If the adaption runs out of its boundaries a CEL (Check Engine Light) will be thrown.

Fuel

Fuel injector valves

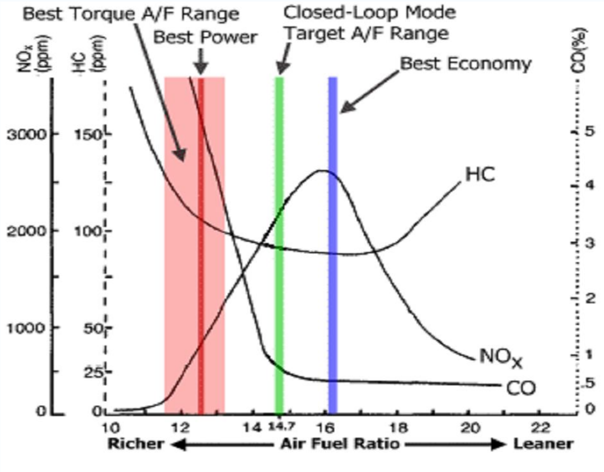
The fuel injector valves are of a solenoid type with needle and seat. They are opened by a current flowing through the injectors coil and are closed by a strong spring when the current is switched off. To ensure as optimal combustion as possible and low exhaust emissions at the same time, the injectors are equipped with four holes, which gives a good distribution of the fuel. The squirts of fuel are very precisely positioned (two jets on the backside on each inlet valve). This puts very high demands on the fixation of the injectors. To secure this fixation the injectors are fixed in pairs by a special retainer between cylinders 1 – 2 and 3 – 4. The injectors are electrically supplied from the main relay, while the ECU grounds the injectors.

Pre injection

When the ignition is switched on, the main relay and fuel pump relay are activated during a few seconds. As soon as the ECU gets the cranking signal (from crankshaft sensor) it initiates a coolant dependant injection with all four injectors simultaneously, this ensures a fast engine start. If the engine is started, and shortly after is switched off again, a new pre injection is initiated after 45 seconds waiting time with the ignition switched off.

Maximum torque

The fuel mixture determines (together with other factors like ignition advance) whether or not the engine can produce its maximum power. The image below shows at what mixtures maximum torque can be achieved and at what mixture maximum economy may be had.



Calculating of injection time

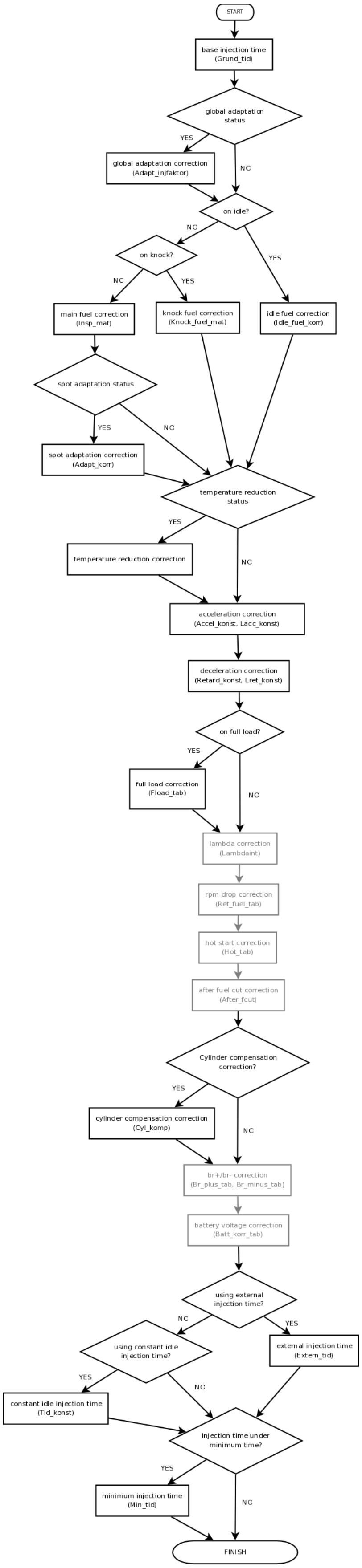
To decide how much fuel needs to be injected into each intake runner the ECU calculates the air mass that has been drawn into the cylinder. The calculation makes use of the cylinder volume (B204 has a displacement of 0,5 liters per cylinder), this cylinder volume holds an equal amount of air which has a density and thus a certain mass. The density is calculated using the absolute pressure and temperature in the intake manifold. The air mass for combustion has now been calculated and this value is divided by 14,7 (stoichiometric relation for gasoline mass to air mass) to determine the required fuel mass for each combustion to inject. Since we know the flow capacity of the injector and the density of the fuel (pre programmed values) the ECU can calculate the duration of the injection.

Using the oxygen sensor (front) the injection duration is corrected so we receive Lambda=1 (stoichiometric combustion). When hard acceleration occurs the lambda correction is masked out and WOT (Wide Open Throttle) enrichment occurs for maximum performance. When opening the throttle, acceleration enrichment occurs and when closing the throttle deceleration enleanment occurs. During a cold start and warm up, before lambda correction is activated, coolant temperature dependent fuel enrichment occurs. With a warm engine and normal battery voltage the duration of injection varies between 2,5 ms at idle and approx. 18 – 20 ms at full torque.
After the calculation for injection time a constant is used to determine the actual time for injector opening time. This constant is also used for example to compensate for different fuel type (e.g. alcohol based fuels). When using standard injectors on a turbo engine (should also be used on a N/A engine but according to EPC there are different injectors) with normal displacement the injection constant is 20-21 in easy mode in T5Suite. The first calculation of injection time is:

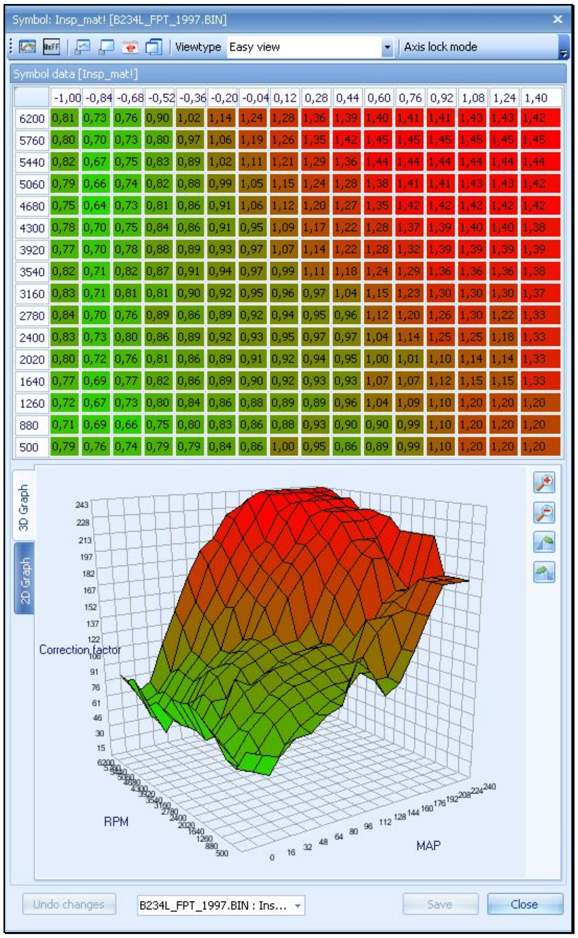
$$Grund\_tid = Inj\_konst * (((Lufttemp\_faktor + 384) * P\_Manifold10) / 512)$$

Finally the volumetric efficiency table is drawn into the equation. This table holds correction values for different manifold pressures and rpm sites. This correction value ranges from ~0.5 to ~1.5. <u>If a different engine is used, or for example a different turbo charger, this map should be adjusted</u>.
Engine VE is determined by the hardware used (also see chapter: Volumetric efficiency).
You can see that the VE curve is steep, this is due that:

$$CorrectedValue = Grund\_tid * ((matrixvalue + 128) / 256)$$


START
base injection time (Grund_tid)
global adaptation status
YES
global adaptation correction (Adapt_injfaktor)
NC
on idle?
NC
on knock?
YES
NC
YES
main fuel correction (Insp_mat)
knock fuel correction (Knock_fuel_mat)
idle fuel correction (Idle_fuel_korr)
spot adaptation status
YES
NC
spot adaptation correction (Adapt_korr)
temperature reduction status
YES
temperature reduction correction
NC
acceleration correction (Accel_konst, Lacc_konst)
deceleration correction (Retard_konst, Lret_konst)
on full load?
YES
full load correction (Fload_tab)
NC
lambda correction (Lambdaint)
rpm drop correction (Ret_fuel_tab)
hot start correction (Hot_tab)
after fuel cut correction (After_fcut)
Cylinder compensation correction?
YES
cylinder compensation correction (Cyl_komp)
NC
br+/br- correction (Br_plus_tab, Br_minus_tab)
battery voltage correction (Batt_korr_tab)
using external injection time?
NC
YES
using constant idle injection time?
external injection time (Extern_tid)
YES
NC
constant idle injection time (Tid_konst)
injection time under minimum time?
YES
minimum injection time (Min_tid)
NC
FINISH

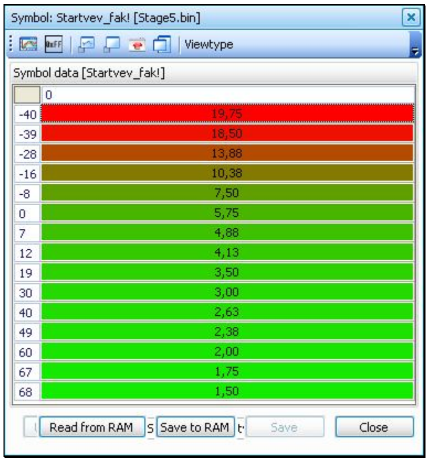
Note that the minimum injection duration is determined by the variable Min_tid!. This variable normally has a value of 250. (1 ms) and that the battery voltage will adjust final duration through table Batt_korr!.



	-1,00	-0,84	-0,68	-0,52	-0,36	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24	1,40
6200	0,81	0,73	0,76	0,90	1,02	1,14	1,24	1,28	1,36	1,39	1,40	1,41	1,41	1,43	1,43	1,42
5760	0,80	0,70	0,73	0,80	0,97	1,06	1,19	1,26	1,35	1,42	1,45	1,45	1,45	1,45	1,45	1,45
5440	0,82	0,67	0,75	0,83	0,89	1,02	1,11	1,21	1,29	1,36	1,44	1,44	1,44	1,44	1,44	1,44
5060	0,79	0,66	0,74	0,82	0,88	0,99	1,05	1,15	1,24	1,28	1,38	1,41	1,41	1,43	1,43	1,42
4680	0,75	0,64	0,73	0,81	0,86	0,91	1,06	1,12	1,20	1,27	1,35	1,42	1,42	1,42	1,42	1,42
4300	0,78	0,70	0,75	0,84	0,86	0,91	0,95	1,09	1,17	1,22	1,28	1,37	1,39	1,40	1,40	1,38
3920	0,77	0,70	0,78	0,88	0,89	0,93	0,97	1,07	1,14	1,22	1,28	1,32	1,39	1,39	1,39	1,39
3540	0,82	0,71	0,82	0,87	0,91	0,94	0,97	0,99	1,11	1,18	1,24	1,29	1,36	1,36	1,36	1,38
3160	0,83	0,71	0,81	0,81	0,90	0,92	0,95	0,96	0,97	1,04	1,15	1,23	1,30	1,30	1,30	1,37
2780	0,84	0,70	0,76	0,89	0,86	0,89	0,92	0,94	0,95	0,96	1,12	1,20	1,26	1,30	1,22	1,33
2400	0,83	0,73	0,80	0,86	0,89	0,92	0,93	0,95	0,97	0,97	1,04	1,14	1,25	1,25	1,18	1,33
2020	0,80	0,72	0,76	0,81	0,86	0,89	0,91	0,92	0,94	0,95	1,00	1,01	1,10	1,14	1,14	1,33
1640	0,77	0,69	0,77	0,82	0,86	0,89	0,90	0,92	0,93	0,93	1,07	1,07	1,12	1,15	1,15	1,33
1260	0,72	0,67	0,73	0,80	0,84	0,86	0,88	0,89	0,89	0,96	1,04	1,09	1,10	1,20	1,20	1,20
880	0,71	0,69	0,66	0,75	0,80	0,83	0,86	0,88	0,93	0,90	0,90	0,99	1,10	1,20	1,20	1,20
500	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99	1,10	1,20	1,20	1,20

Cranking enrichment

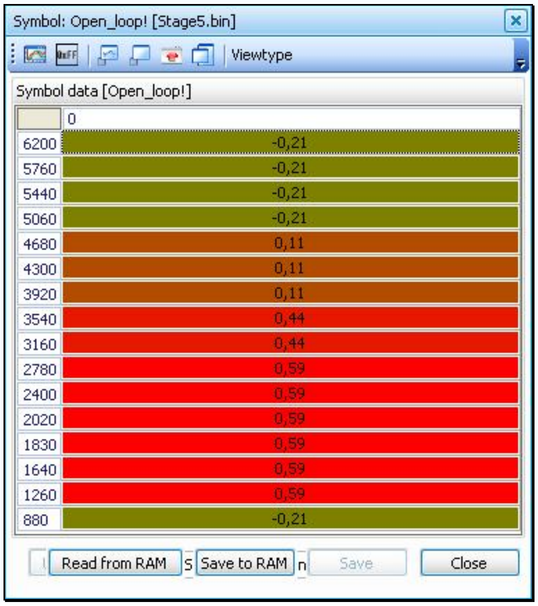
To facilitate a better start cranking enrichment is needed. The factor that compensates which is coolant temperature dependant is stored in Startvev_fak! (what's in a name).


Symbol: Startvev_fak! [Stage5.bin]
Viewtype
Symbol data [Startvev_fak!]
Read from RAM
Save to RAM
Save
Close

	0
-40	19,75
-39	18,50
-28	13,88
-16	10,38
-8	7,50
0	5,75
7	4,88
12	4,13
19	3,50
30	3,00
40	2,63
49	2,38
60	2,00
67	1,75
68	1,50

Lambda correction

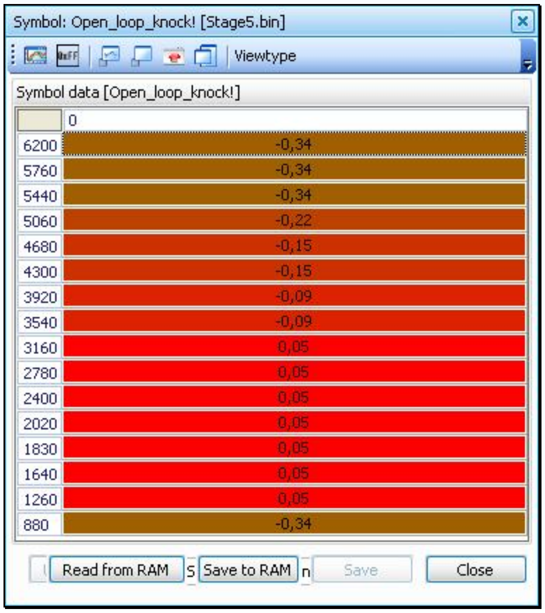
The catalyst requires that the fuel/air mixture is stoichiometric. This means that the mixture is neither rich nor lean, it is exactly 14,7 kg air to 1 kg gasoline (Lambda=1). That is why the system is equipped with an oxygen sensor in the forward part of the exhaust system. The sensor is connected to pin 23 in the ECU and is grounded in the ECU via pin 47. The exhaust fumes pass the oxygen sensor. The content of oxygen in the exhaust fumes is measured through a chemical reaction, this results in an output voltage. If the engine runs rich (Lambda lower than 1) the output voltage would be approx. 0,9 V and if the engine runs lean (Lambda higher than 1) the output voltage would be 0,1 V. The output voltage swings when Lambda passes 1. The ECU continuously corrects the injection duration so that Lambda=1 is always met. To be able to function, the oxygen sensor needs to be hot, this requirement is met by electrically preheating the sensor. The preheating element is fed by B+ via fuse 38 and the main relay, the sensor is grounded in the ECU via pin 50. The ECU estimates the temperature on the exhaust gases (EGT) based on engine load and engine speed (RPM). At high EGT the electrical pre heating is disconnected. The lambda correction is masked during the engines first 640 revolutions after start if the coolant temperature exceeds 18°C (64F) at load ranges over idle and under WOT, or 32°C (90F) at idle. Operation can be divided into two parts, open loop mode and closed loop mode. In closed loop mode the feedback signal from the lambda sensor is used to keep the mixture stoichiometric (approx. 14.7 : 1) while in open loop mode the system only depends on the maps, no feedback from the lambda sensor is used to compensate for lean or rich mixture. The point where the system switches from closed to open loop mode is determined by a couple of maps of which the most important one is: Open_loop! This map indicates the load levels for each rpm site from which the system should run in open loop mode. This is also indicated in T5Suite in the main injection map and the fuel knock map. The load levels in knock mode are different from the ones for normal mode operation and are stored in Open_loop_knock!



Symbol: Open_loop! [Stage5.bin]

Symbol data [Open_loop!]

	0
6200	-0,21
5760	-0,21
5440	-0,21
5060	-0,21
4680	0,11
4300	0,11
3920	0,11
3540	0,44
3160	0,44
2780	0,59
2400	0,59
2020	0,59
1830	0,59
1640	0,59
1260	0,59
880	-0,21



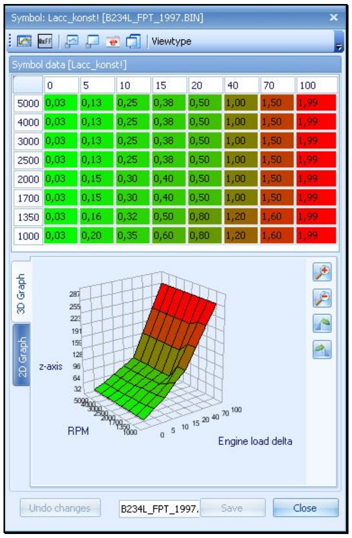
Symbol: Open_loop_knock! [Stage5.bin]

Symbol data [Open_loop_knock!]

	0
6200	-0,34
5760	-0,34
5440	-0,34
5060	-0,22
4680	-0,15
4300	-0,15
3920	-0,09
3540	-0,09
3160	0,05
2780	0,05
2400	0,05
2020	0,05
1830	0,05
1640	0,05
1260	0,05
880	-0,34

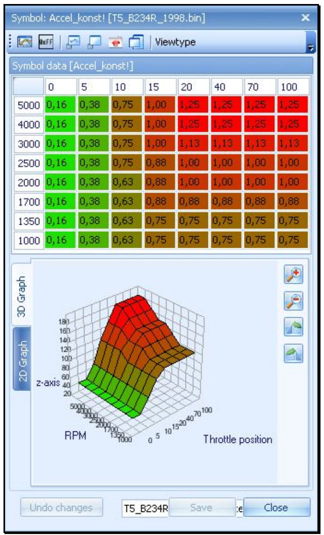
Acceleration and deceleration

When going into transients for pedal position the throttle opening changes fast. Suppose we floor the pedal suddenly while cruising at 2500 rpm. The sudden opening of the throttle draws much more air into the engine but because of the sudden 'rush' the fuel which is hanging in the air cannot follow in the same rate. Hence, the mixture goes lean even if we supply enough fuel. This is why we need acceleration enrichment (and deceleration enleanment). When pressing down or releasing the accelerator pedal Trionic tries to maintain Lambda at 1. This is done by fetching values from an acceleration or deceleration table (Lacc_konst, Accel_konst, Lret_konst and Retard_konst).
When the accelerator pedal is pressed down or engine load increases the system checks both acceleration tables (Lacc_konst and Accel_konst). Lacc_konst is a function of engine load while Accel_konst is a function of throttle position (how far the throttle body is opened and hence how much air is entering the engine). Both tables result in an enrichment factor. The enrichment factor that is the largest is the one that will be applied. The same goes for enleanment. Whenever the accelerator pedal is released or engine load decreases the system checks both Lret_konst and Retard_konst. Lret_konst is a function of engine load and Retard_konst is a function of throttle position. Both tables result in an enleanment factor. The largest value is the value that will be applied.



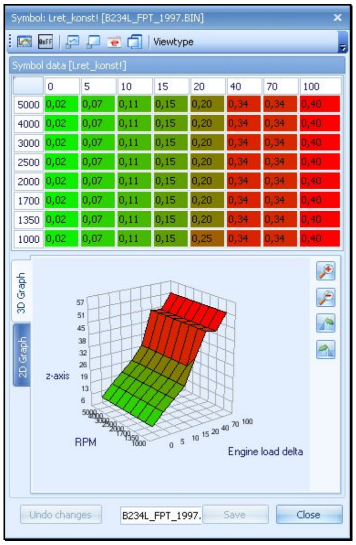
Symbol data [Lacc_konst!]

	0	5	10	15	20	40	70	100
5000	0,03	0,13	0,25	0,38	0,50	1,00	1,50	1,99
4000	0,03	0,13	0,25	0,38	0,50	1,00	1,50	1,99
3000	0,03	0,13	0,25	0,38	0,50	1,00	1,50	1,99
2500	0,03	0,13	0,25	0,38	0,50	1,00	1,50	1,99
2000	0,03	0,15	0,30	0,40	0,50	1,00	1,50	1,99
1700	0,03	0,15	0,30	0,40	0,50	1,00	1,50	1,99
1350	0,03	0,16	0,32	0,50	0,80	1,20	1,60	1,99
1000	0,03	0,20	0,35	0,60	0,80	1,20	1,60	1,99



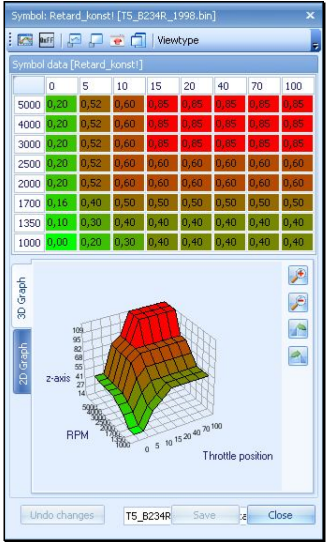
Symbol data [Accel_konst!]

	0	5	10	15	20	40	70	100
5000	0,16	0,38	0,75	1,00	1,25	1,25	1,25	1,25
4000	0,16	0,38	0,75	1,00	1,25	1,25	1,25	1,25
3000	0,16	0,38	0,75	1,00	1,13	1,13	1,13	1,13
2500	0,16	0,38	0,75	0,88	1,00	1,00	1,00	1,00
2000	0,16	0,38	0,63	0,88	1,00	1,00	1,00	1,00
1700	0,16	0,38	0,63	0,88	0,88	0,88	0,88	0,88
1350	0,16	0,38	0,63	0,75	0,75	0,75	0,75	0,75
1000	0,16	0,38	0,63	0,75	0,75	0,75	0,75	0,75



Symbol data [Lret_konst!]

	0	5	10	15	20	40	70	100
5000	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
4000	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
3000	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
2500	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
2000	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
1700	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
1350	0,02	0,07	0,11	0,15	0,20	0,34	0,34	0,40
1000	0,02	0,07	0,11	0,15	0,25	0,34	0,34	0,40

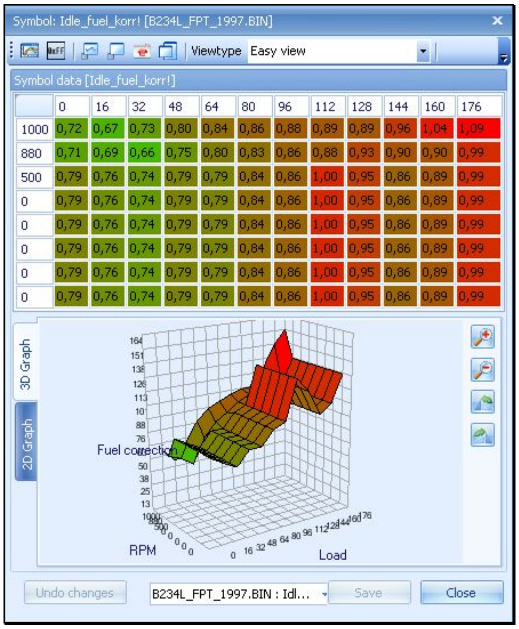


Symbol data [Retard_konst!]

	0	5	10	15	20	40	70	100
5000	0,20	0,52	0,60	0,85	0,85	0,85	0,85	0,85
4000	0,20	0,52	0,60	0,85	0,85	0,85	0,85	0,85
3000	0,20	0,52	0,60	0,85	0,85	0,85	0,85	0,85
2500	0,20	0,52	0,60	0,60	0,60	0,60	0,60	0,60
2000	0,20	0,52	0,60	0,60	0,60	0,60	0,60	0,60
1700	0,16	0,40	0,50	0,50	0,50	0,50	0,50	0,50
1350	0,10	0,30	0,40	0,40	0,40	0,40	0,40	0,40
1000	0,00	0,20	0,30	0,40	0,40	0,40	0,40	0,40

Idle correction

To be able to maintain a steady idle state a separate idle fuel map is used (Idle_fuel_korr). This table aids in keeping the idle rpm smooth by correcting fuel injection a bit.



Symbol: Idle_fuel_korr! [B234L_FPT_1997.BIN]

Viewtype: Easy view

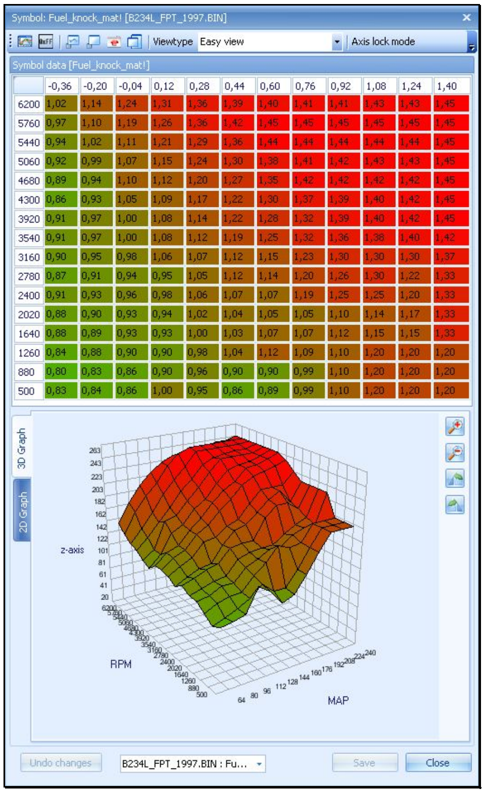
Symbol data [Idle_fuel_korr!]

	0	16	32	48	64	80	96	112	128	144	160	176
1000	0,72	0,67	0,73	0,80	0,84	0,86	0,88	0,89	0,89	0,96	1,04	1,09
880	0,71	0,69	0,66	0,75	0,80	0,83	0,86	0,88	0,93	0,90	0,90	0,99
500	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99
0	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99
0	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99
0	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99
0	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99
0	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99

B234L_FPT_1997.BIN : Idl...

Knocking

When knocking occurs inside the engine (sensed through the DI cartridge) evenly in all cylinders a tiny bit of extra fuel will be injected. When the manifold pressure (MAP) is over 140 kPa (0.4 bar) the normal injection map will be overruled by another map: the knocking fuel map. This map enriches the fuel mixture so that the engine cools down (extra fuel means cooler mixture). If the knocking map kicks in, something is actually wrong in the rest of the mapping (ignition, injection etc). Try to prevent entering the knocking map at all times!



Symbol: Fuel_knock_mat! [B234L_FPT_1997.BIN]

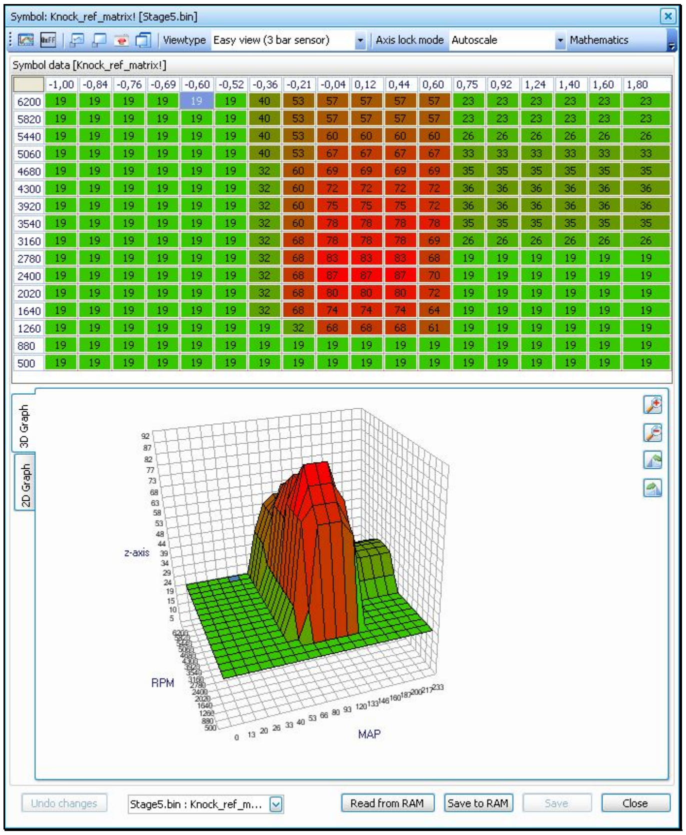
Viewtype: Easy view

Symbol data [Fuel_knock_mat!]

	-0,36	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24	1,40
6200	1,02	1,14	1,24	1,31	1,36	1,39	1,40	1,41	1,41	1,43	1,43	1,45
5760	0,97	1,10	1,19	1,26	1,36	1,42	1,45	1,45	1,45	1,45	1,45	1,45
5440	0,94	1,02	1,11	1,21	1,29	1,36	1,44	1,44	1,44	1,44	1,44	1,45
5060	0,92	0,99	1,07	1,15	1,24	1,30	1,38	1,41	1,42	1,43	1,43	1,45
4680	0,89	0,94	1,10	1,12	1,20	1,27	1,35	1,42	1,42	1,42	1,42	1,45
4300	0,86	0,93	1,05	1,09	1,17	1,22	1,30	1,37	1,39	1,40	1,42	1,45
3920	0,91	0,97	1,00	1,08	1,14	1,22	1,28	1,32	1,39	1,40	1,42	1,45
3540	0,91	0,97	1,00	1,08	1,12	1,19	1,25	1,32	1,36	1,38	1,40	1,42
3160	0,90	0,95	0,98	1,06	1,07	1,12	1,15	1,23	1,30	1,30	1,30	1,37
2780	0,87	0,91	0,94	0,95	1,05	1,12	1,14	1,20	1,26	1,30	1,22	1,33
2400	0,91	0,93	0,96	0,98	1,06	1,07	1,07	1,19	1,25	1,25	1,20	1,33
2020	0,88	0,90	0,93	0,94	1,02	1,04	1,05	1,05	1,10	1,14	1,17	1,33
1640	0,88	0,89	0,93	0,93	1,00	1,03	1,07	1,07	1,12	1,15	1,15	1,33
1260	0,84	0,88	0,90	0,90	0,98	1,04	1,12	1,09	1,10	1,20	1,20	1,20
880	0,80	0,83	0,86	0,90	0,96	0,90	0,90	0,99	1,10	1,20	1,20	1,20
500	0,83	0,84	0,86	1,00	0,95	0,86	0,89	0,99	1,10	1,20	1,20	1,20

Knock sensitivity

For Trionic to be able to determine what received knock signals from the DI cassette are actually knock and which are not, it looks up reference signals from several maps. The most important one is Knock_ref_matrix! shown below.



Viewtype Easy view (3 bar sensor)

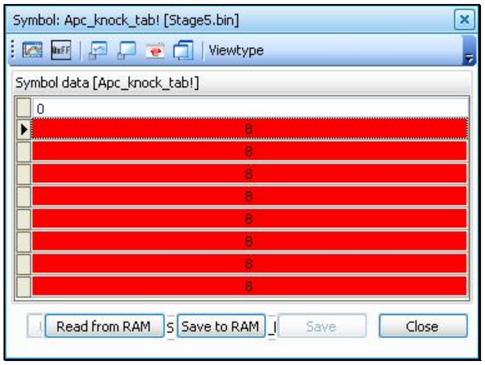
Axis lock mode Autoscale

Stage5.bin : Knock_ref_m...

Symbol data [Knock_ref_matrix!]

	-1,00	-0,84	-0,76	-0,69	-0,60	-0,52	-0,36	-0,21	-0,04	0,12	0,44	0,60	0,75	0,92	1,24	1,40	1,60	1,80
6200	19	19	19	19	19	19	40	53	57	57	57	57	23	23	23	23	23	23
5820	19	19	19	19	19	19	40	53	57	57	57	57	23	23	23	23	23	23
5440	19	19	19	19	19	19	40	53	60	60	60	60	26	26	26	26	26	26
5060	19	19	19	19	19	19	40	53	67	67	67	67	33	33	33	33	33	33
4680	19	19	19	19	19	19	32	60	69	69	69	69	35	35	35	35	35	35
4300	19	19	19	19	19	19	32	60	72	72	72	72	36	36	36	36	36	36
3920	19	19	19	19	19	19	32	60	75	75	75	72	36	36	36	36	36	36
3540	19	19	19	19	19	19	32	60	78	78	78	78	35	35	35	35	35	35
3160	19	19	19	19	19	19	32	68	78	78	78	69	26	26	26	26	26	26
2780	19	19	19	19	19	19	32	68	83	83	83	68	19	19	19	19	19	19
2400	19	19	19	19	19	19	32	68	87	87	87	70	19	19	19	19	19	19
2020	19	19	19	19	19	19	32	68	80	80	80	72	19	19	19	19	19	19
1640	19	19	19	19	19	19	32	68	74	74	74	64	19	19	19	19	19	19
1260	19	19	19	19	19	19	19	32	68	68	68	61	19	19	19	19	19	19
880	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
500	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19

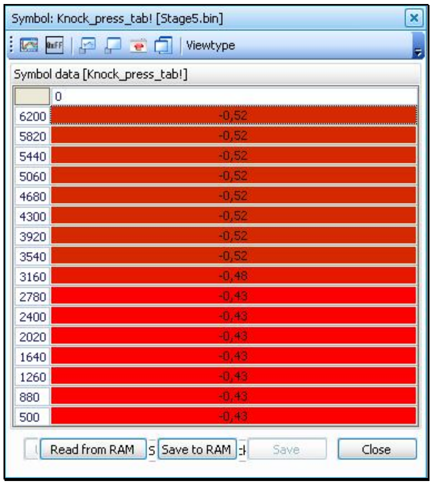
If knock is detected and adding extra fuel and pulling ignition timing doesn't help, Trionic starts to decrease boost. The steps in which this is done are listed in the map " Apc_knock_tab!". In the map shown below all values are 8 which means that every increase will pull 0.08 bar of boost pressure. The current value for this (boost reduction) is stored in Apc_decrese which can be monitored using the CAN bus connection.



Symbol data [Apc_knock_tab!]

0
8
8
8
8
8
8
8
8

Knock detection is only active above a preset level of boost. These levels are stored in Knock_press_tab!.



Symbol: Knock_press_tab! [Stage5.bin]

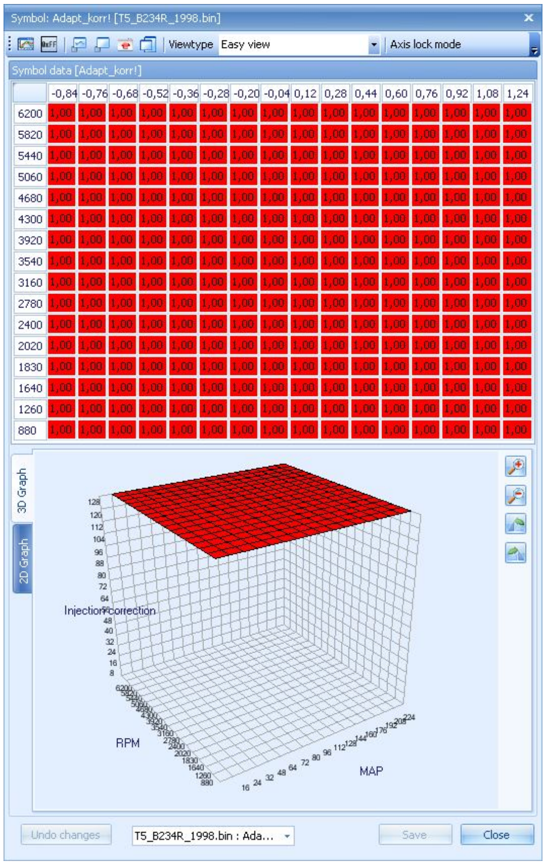
Symbol data [Knock_press_tab!]

	0
6200	-0,52
5820	-0,52
5440	-0,52
5060	-0,52
4680	-0,52
4300	-0,52
3920	-0,52
3540	-0,52
3160	-0,48
2780	-0,43
2400	-0,43
2020	-0,43
1640	-0,43
1260	-0,43
880	-0,43
500	-0,43

Altering VE map

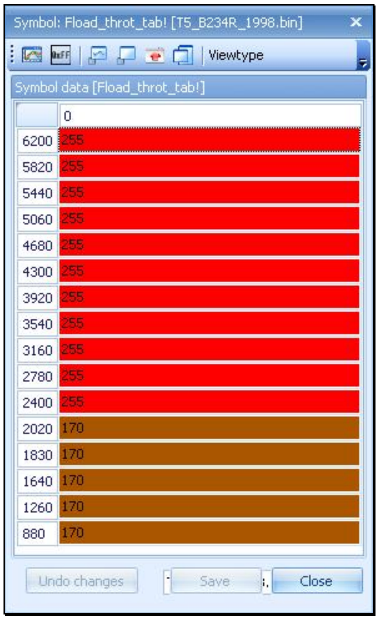
To tune the ECU regarding the fuel injection you must install a wide band lambda sensor to make sure that you hit your desired lambda value when under WOT (Wide Open Throttle). You can make use of an EGT meter here to verify that your desired WOT lambda is enough to keep the EGT below: 1795° F (~980° C) on the SAAB T5 engine. When using alcohol fuel the EGT seldom reaches these temperatures. This means that by using a 3" down pipe and a fairly sized turbine the EGT meter can be redundant.

The VE-map must be tuned to make the engine run at lambda 1 in all load/RPM points that don't need a richer mixture. In wide open throttle situations an AFR of 12 or even a bit richer are required. This job is already done by the ECU when it regulates the fuel injection in respect of lambda and storing the result in the Adapt_korr!. To be able to use adaption maps you must be able to pull this data from your ECU without disconnecting the power and use it to adapt the initial VE map (see chapter CAN Bus).

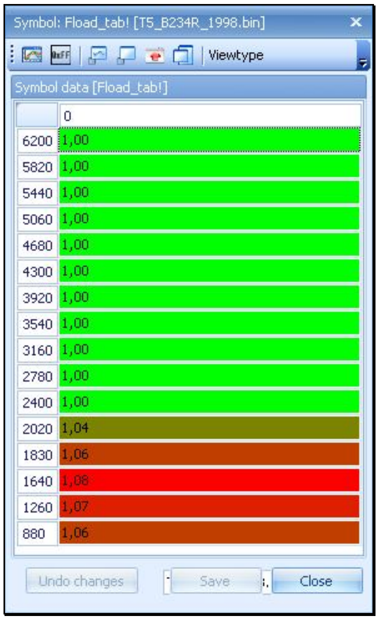


	-0,84	-0,76	-0,68	-0,52	-0,36	-0,28	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24
6200	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5820	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5440	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
5060	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4680	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
4300	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3920	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3540	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3160	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2780	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2400	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2020	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1830	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1640	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1260	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
880	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

When the engine is run at WOT conditions, Fload_throttle_tab! indicates when the throttle is considered to be wide open and full load enrichment is activated. The WOT enrichment map Fload_tab! is used to add more fuel to the calculated amount of fuel to be injected. Note the interesting original Fload_tab! Over 2400 RPM the lambda is masked when the throttle is fully pressed down but there is no additional fuel than what is allowed from the VE matrix. Below 2400 RPM the lambda is masked and extra fuel is injected. Here are the right tables for adding fuel to cool your exhaust gases.


Symbol: Fload_throt_tab! [T5_B234R_1998.bin]
Viewtype
Symbol data [Fload_throt_tab!]
Undo changes
Save
Close

	0
6200	255
5820	255
5440	255
5060	255
4680	255
4300	255
3920	255
3540	255
3160	255
2780	255
2400	255
2020	170
1830	170
1640	170
1260	170
880	170



	0
6200	1,00
5820	1,00
5440	1,00
5060	1,00
4680	1,00
4300	1,00
3920	1,00
3540	1,00
3160	1,00
2780	1,00
2400	1,00
2020	1,04
1830	1,06
1640	1,08
1260	1,07
880	1,06

The values shown in Fload_throt_tab are throttle position sensor values and in Fload_tab the values are values to add to the fuel calculation.
The EGT can be allowed for up to 1795° F (~980° C), it is up to you how you will be using it. A quarter mile (402 m) dragster can be allowed for hitting that temperature after about 11 seconds. A more useful temperature will be ~1652° F (900° C) at WOT when every day driving. If you must go below AFR 11 when driving on gasoline to prevent a high EGT something is wrong. Upgrade your turbine and add a 3" down pipe, the EGT rises if there is a bottleneck in the passage for the exhaust gases or if the ignition is too late (not enough advance).

When tuning the fuel maps make sure that you are tuning the right maps. When accelerating and decelerating there are enrichment and leaning tables that starts to function to let you hit Lambda 1. But between idle and 2400 RPM this is overtaken by the WOT enrichment to get maximum power without pinging when accelerating. The downside is that the catalyst stops to oxidise HC and CO fractions in the exhaust gases. If you are over 140 kPa you could be running on the knock map (Fuel_knock_mat!) because the engine had started to knock. When this happens the engine should stop accelerating due to the simultaneous change of ignition map (Ign_map_2). Further tuning of the normal map is then useless. This could be indicated that the boost request map must be altered to "ridiculous" levels to make the engine continue to accelerate. A second point of view is that the knock matrix must be changed according to the changes made in the normal VE-table, strive to keep the proportional differences of the maps.
Always make sanity checks of what you are doing so that your engine can deliver maximum power without excessive emissions and with a minimum risk of mechanical failures.

Finally, give thought to the engine not being lean when cold by: verifying that the normal (stock) warm up enrichments are enough, using OBD II or the Canbus to check the short term fuel trim during warm up to a coolant temperature of 85° C.

A short list of what has been discussed:

- Always strive for Lambda 1 when using the engine in normal traffic.
- Let the ECU do the boring job for you.
- Keep your EGT with margins for overheating the turbine.
- Make changes in all necessary maps but think of the WOT maps.
- Don't forget the Fuel_knock_mat!
- Don't let the engine run lean.

Adaption

The ECU calculates the injection duration based on MAP and intake temperature. Injection duration is then corrected by multiplication of a correction factor, which is fetched from the main fuel matrix (Insp_mat!) which depends on MAP and RPM. The need to correct the injection duration is due to the fact that the volumetric efficiency of the engine is dependent on the engine speed (RPM). The last correction is made with the lambda correction, this results in a stoichiometric combustion (Lambda=1). The lambda correction is allowed to adjust the calculated injection duration by ±25 %. The ECU can change the correction factors in the main fuel matrix based on the lambda correction. This ensures good drivability, fuel consumption and emissions when lambda correction isn't activated. This is called Adaption.

Pointed adaption

If the ECU calculates the injection duration to be 8 ms but the lambda correction adjusts it to 9 ms due low fuel pressure, the ECU will "learn" the new injection duration. This is done by changing the correction factor for that particular RPM and load point in the main fuel matrix to a new correction factor resulting in 9 ms injection duration. The correction factor in this example will be raised by 9/8 (+12 %). The pointed adaption can change the points in the main fuel matrix by ±25 %. Adaption occurs every fifth minute and takes 30 seconds to finish, the criteria for the adaption are: Lambda correction is activated and the coolant temperature is above 64°C (147F). During the adaption the ventilation valve on the carbon canister is held closed.

Global adaption

The global adaption on OBDII variants occurs during driving. On non OBDII variants the global adaption occurs 15 minutes after engine shut down. When the engine is inside a defined load and RPM range (60 – 120 kPa and 2000 – 3000 RPM) no pointed adaption will occur. Instead, all points in the fuel matrix will be changed by a multiplication factor. Global adaption can change the points in the main fuel matrix by ±25 % (Tech2 shows ±100 %). Adaption occurs every fifth minute and takes 30 seconds to finish, the criteria for the adaption are: Lambda correction is activated and the coolant temperature is above 64°C (147F). During the adaption the ventilation valve on the carbon canister is held closed.

Fuel save

With fully closed throttle and engine RPM over 1900 RPM when in third, fourth and fifth gear, a fuel shutoff will occur after a small delay. On automobiles with automatic transmission fuel shutoff is active in all stages. The injectors are reactivated when the RPM hits 1400 RPM.

Converting to E85 fuel

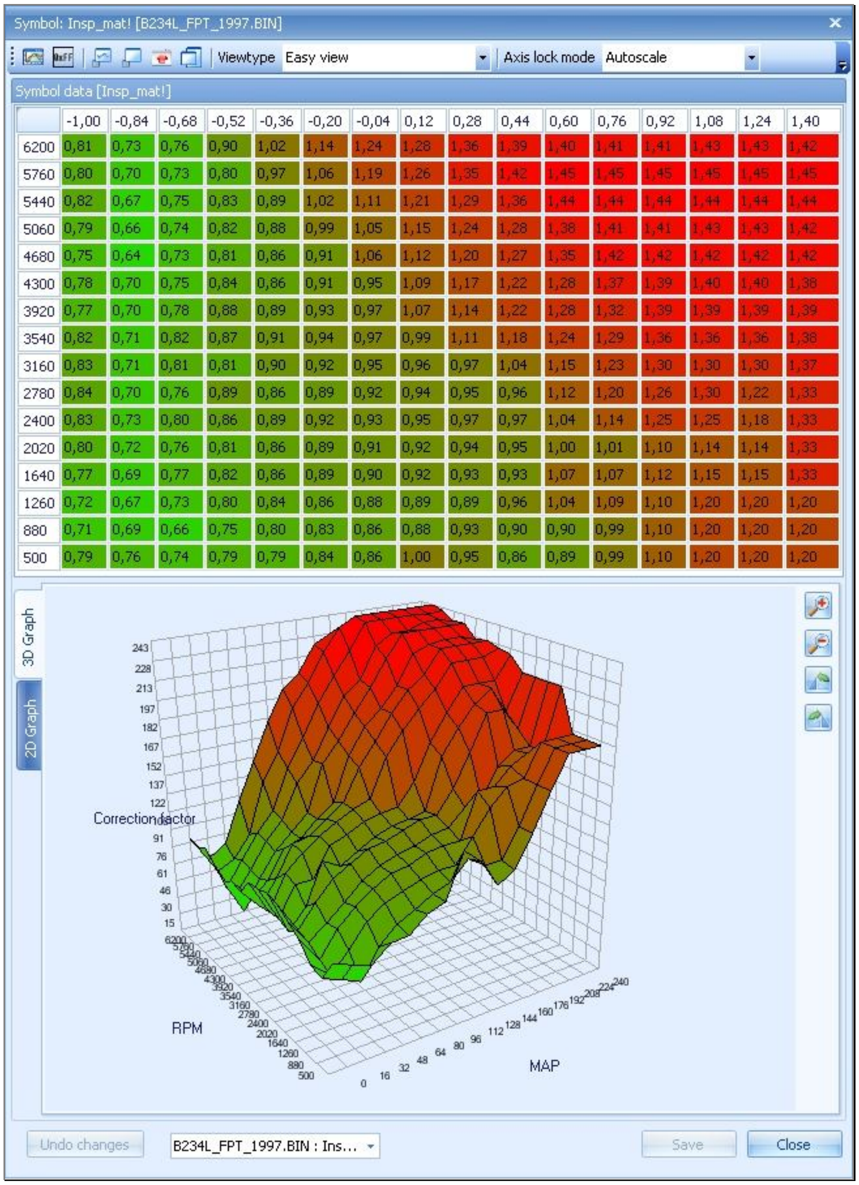
If you want to setup your car for E85 fuel use, you will need to inject 30% - 40% more fuel. Change the inj_konst! to reach the 30% to 40 % more fuel needed for E85 regardless of what injectors you are currently using. If the injectors that you plan to mount are flowing the desired percentage of fuel more, you will not need to change the Inj_konst!

The Ethanol fuel E85 has different combustion characteristics resulting in more ignition timing advance at low RPM´s and less ignition advance at high RPM´s compared to gasoline. This is the result of the higher combustion rate of Ethanol and its less initial combustion rate. I.e. it's harder to ignite but when it ignites it burns faster than gasoline. The main error E85 converters do is, advance the timing to restore the torque and not retard the ignition later when needed, thereby putting unnecessary stress to the piston by passing MBT.

Additional information can be found on http://etanol.nu/forumrecover/index.php

Fuel consumption measurement

The wire from the ECU to the third injector is also connected to the main instrument cluster. The main instrument calculates the fuel consumption based on the injection pulses duration. The fuel consumption is used to help getting an accurate presentation of the fuel level in the fuel tank and to calculate average fuel consumption in the SID.



Symbol: Insp_mat! [B234L_FPT_1997.BIN]

Viewtype Easy view

Axis lock mode Autoscale

Symbol data [Insp_mat!]

	-1,00	-0,84	-0,68	-0,52	-0,36	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24	1,40
6200	0,81	0,73	0,76	0,90	1,02	1,14	1,24	1,28	1,36	1,39	1,40	1,41	1,41	1,43	1,43	1,42
5760	0,80	0,70	0,73	0,80	0,97	1,06	1,19	1,26	1,35	1,42	1,45	1,45	1,45	1,45	1,45	1,45
5440	0,82	0,67	0,75	0,83	0,89	1,02	1,11	1,21	1,29	1,36	1,44	1,44	1,44	1,44	1,44	1,44
5060	0,79	0,66	0,74	0,82	0,88	0,99	1,05	1,15	1,24	1,28	1,38	1,41	1,41	1,43	1,43	1,42
4680	0,75	0,64	0,73	0,81	0,86	0,91	1,06	1,12	1,20	1,27	1,35	1,42	1,42	1,42	1,42	1,42
4300	0,78	0,70	0,75	0,84	0,86	0,91	0,95	1,09	1,17	1,22	1,28	1,37	1,39	1,40	1,40	1,38
3920	0,77	0,70	0,78	0,88	0,89	0,93	0,97	1,07	1,14	1,22	1,28	1,32	1,39	1,39	1,39	1,39
3540	0,82	0,71	0,82	0,87	0,91	0,94	0,97	0,99	1,11	1,18	1,24	1,29	1,36	1,36	1,36	1,38
3160	0,83	0,71	0,81	0,81	0,90	0,92	0,95	0,96	0,97	1,04	1,15	1,23	1,30	1,30	1,30	1,37
2780	0,84	0,70	0,76	0,89	0,86	0,89	0,92	0,94	0,95	0,96	1,12	1,20	1,26	1,30	1,22	1,33
2400	0,83	0,73	0,80	0,86	0,89	0,92	0,93	0,95	0,97	0,97	1,04	1,14	1,25	1,25	1,18	1,33
2020	0,80	0,72	0,76	0,81	0,86	0,89	0,91	0,92	0,94	0,95	1,00	1,01	1,10	1,14	1,14	1,33
1640	0,77	0,69	0,77	0,82	0,86	0,89	0,90	0,92	0,93	0,93	1,07	1,07	1,12	1,15	1,15	1,33
1260	0,72	0,67	0,73	0,80	0,84	0,86	0,88	0,89	0,89	0,96	1,04	1,09	1,10	1,20	1,20	1,20
880	0,71	0,69	0,66	0,75	0,80	0,83	0,86	0,88	0,93	0,90	0,90	0,99	1,10	1,20	1,20	1,20
500	0,79	0,76	0,74	0,79	0,79	0,84	0,86	1,00	0,95	0,86	0,89	0,99	1,10	1,20	1,20	1,20

Image 7: Stock fuel injection table (9000 FPT 200bhp T25 turbo)

HOWTO: Get your fuelling under control

If you start tuning for different turbo models, different injectors or anything else that changes either the air path or the fuelling path you will need to remap the fuelling part in Trionic 5. Trionic 5 will adapt for approx. 25% in fuelling but you don't want to be on the end of adaption after your changes of course.

Targets for mapping

What to map for? That is the primary question to start with. Normally, off-boost and a little but on-boost the system will be running in closed loop (lambda signal is taken into account when calculating optimum mixture) and the ECUs target will always be 14.7 AFR (lambda = 1). The actual mixture will swing a little bit around this target because of the usage of a narrowband lambda sensor (14.x-15.x). When you go on-boost the system will leave closed loop mode and enter open-loop mode. In this mode the lambda signal is not used and the system depends only on the fuelling information in the maps that it has. On full load you will want to have a target of approx. AFR 11.x. Maximum torque is made at 11.1 AFR but since the curve drops off pretty steep when going richer most tuners aim for 11.5-12.0 at full load. The path towards full load (on-boost but still not full load) can gradually go from 14.x to 11.x. Of course you will need a calibrated wideband lambda sensor for mapping for optimum mixtures.

Main fueling constant [inj_konst]

To correct the system for different injectors a injector constant variable is present. It is called Inj_konst! And is around a value of 20 with stock injectors. If you go for larger injectors this constant will have to be dropped because the injectors deliver more fuel per millisecond. Changing from stock to for example 630cc injectors will drop the constant from 20 to 14-16 range. You can also use this constant to get a little more tuning headroom in the main fuel map since that is mostly mapped to the top (maxed out). If you increase the injector constant a slightly you can drop the main fuel map a bit to gain more control in the upper regions of this map.

Main fuel injection map [insp_mat]

This is a map that determines the corrections to be made at each rpm/load point to the pre-calculated injection duration. It ranges from 0.5 to 1.5 and the value is a multiplier to the injection duration that was calculated based on manifold pressure, IAT and coolant already. With this map you can map out flat spots and make sure that every rpm/load site sits at YOUR desired AFR value.

Knock fuel injection map [fuel_knock_mat]

If the system detects knock it will inject a different amount of fuel. This fuel amount is no longer calculated using the main fuel injection map but it takes the correction factor from this map. So, always make sure that this map has higher values (5-10%) then the main fuel map.

Open loop vs. closed loop [open_loop]

If you want the system to leave closed loop at different rpm/load point you can alter the settings in this map. So, if you want to run different AFR values for a rpm/load point that is actually in the closed loop range (these ranges are indicated in T5Suite by marking the cells in the main fuel map) and would thus be 14.7, you will have to alter the open_loop map. In this way you can determine the AFR value by changing the values in the main fuel map. Note that in knocking conditions a different map is used. You can also switch off closed loop as a whole by changing the firmware settings in the system. This way you gain full control of the fuelling part but lose the system's ability to adapt for fuel pump wear etc.

Coldstart enrichment [Eftersta_fak*]

To have the car run properly when the engine is cold (vaporization is worse) it needs to inject more fuel. These values are coolant temperature dependant so they can also be used for semi-cold start (e.g. coolant = 50 degrees). The values in the maps are also correction factor like they are in the main fuelling map.

Acceleration/deceleration enrichment

If you suddenly open or close the throttle plate the air flow changes dramatically in the intake. Since air is lighter (weighs less) than petrol vapor/droplets the air will react faster than the petrol in it. Opening the throttle will therefore result in a lean mixture because the petrol will lag behind on the air. Closing the throttle will cause the air to stop faster than the petrol because mass reacts slower and hence the mixture will go too rich. To compensate for these phenomena the acceleration and deceleration maps are there to try to keep the desired AFR when changing throttle or changing load (which amounts to the same actually because load change is airflow change). Maps are Accel_konst, Lacc_konst, Lret_konst and Retard_konst.

Battery correction for injectors [Batt_korr_tab]

Different battery voltages result in different injection durations because the injector opening time varies with the voltage applied to it. Lower voltage means that the injector valve takes longer to open and hence less fuel is injected in the same injector duration. This calls for correction factors that depend on battery voltage.

Watch out for adaption

When reading the above you might think that the closed loop values in the main fuel map are redundant because the system will use lambda feedback anyway. This is not true because the system will adjust the adaption trim based on the feedback values from the lambda sensor compared to the values within the closed loop range of the main fuel map. If you map the closed loop part incorrectly the system will react by adjusting the adaption values (both long term and point adaption – per cell that is). The horror is in the fact that long term trim is also applied to WOT (wide open throttle, so full load) conditions. The you map closed loop too rich, the system will trim the fuel down using the long term trim and your WOT mixture will also go leaner… watch out for this!

Watch out for LPG/petrol combinations while tuning

One of the main problems with running LPG is that the Trionic adaption system will also adapt when you are running LPG. Most people tend to run on LPG most of the time, only running petrol on engine start and when LPG runs out. This means that adaption is done mostly on LPG. If the LPG manufacturer installs and calibrates the system he will set a multiplier in the LPG computer to correct for the extra amount of fuel needed when running LPG. If he does this incorrectly – for example he enters a number a bit too big – the fuelling will be a little off when running LPG. Trionic will adapt to this – in the example it will trim the fuel down (cases are known that Trionic has to adapt so much that the CEL comes on with adaption range errors). No problem when still running on LPG but when adaption is done (lets assume 10% off) and the driver switches to petrol the engine WILL be running lean. Indicators are that the engine will not run properly on coldstart (it will start on petrol with the 10% off and thus will not inject the intended amount of fuel).

Ignition

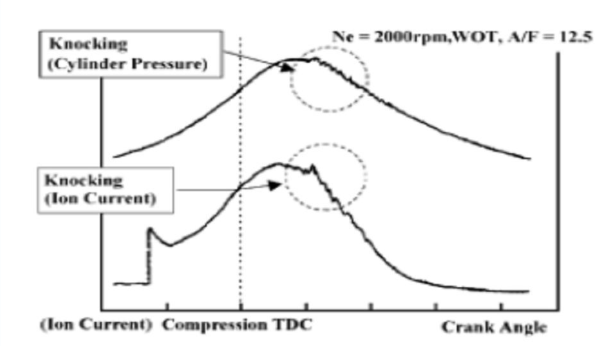
Ignition cassette

The ignition cassette is mounted on the valve cover on top of the spark plugs. The ignition cassette houses four ignition coils/transformers whose secondary coil is direct 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. To the ignition cassette there are four triggering lines connected 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 ignition cassettes B+ intake) 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.

Ignition regulation

At start the ignition point is 10° BTDC. To facilitate start when coolant temperature is below 0°C the ECU will ground each trigger line 210 times/second between 10° BTDC and 20° ATDC, at which a “multi spark” will appear. The function is active up to an engine speed of 900 RPM. At idle a special ignition matrix is utilized. Normal ignition point is 6°-8° BTDC. If the engine stalls e.g. cooling fan activation the ignition point is advanced up to 20 ° BTDC in order to increase the engines torque to restore the idle RPM. In the same way the ignition is retarded if the engines RPM is increased. When the TPS senses an increase in throttle opening the ECU leaves the idle ignition timing map and regulates the ignition timing in respect of load and engine speed.

During engine operations the Ignition cassette continuously monitors the ion currents in the cylinders and sends a signal to the Trionic ECU on pin 44, in the event of knocking. The logic for this function rests solely in the ignition cassette and is adaptive to be able to handle disturbing fuel additives. The Trionic ECU is well aware of which cylinder that has ignited and could hence cope with the information fed through one pin. The signal to pin 44 and ion current in the combustion chamber is related to each other. When this signal reaches a certain level the ECU interprets this as a knocking event and firstly lowers the ignition advance by 1.5° on this cylinder. If the knocking is repeated the ignition advance is lowered further by 1.5 °, up to 12°. In case of the same lowering of the ignition timing advance in all cylinders the ECU adds a small amount of fuel to all cylinders. If knocking occurs when the MAP is over 140 kPa the knocking is regulated by switching both fuel injection matrix and ignition advance matrix. If this is not sufficient the charging pressure is lowered. This purpose of this procedure is to maintain good performance. If the signal between the ignition cassette and the ECU is lost, the charging pressure is lowered to basic charging pressure and the ignition timing advance is retarded by 12° if a risk of knocking exists due to engine load. Refer to appendix VII for more information about knock detection.



Knocking detection

Combustion signals

The Trionic system lacks a camshaft position sensor. This sensor is normally a prerequisite for a sequential pre ignition/pinging regulation and fuel injection. Saab Trionic must decide whether cylinder one or cylinder four ignites when the crank shaft position sensor indicates that cylinder one and four is at TDC. This is done by the help of ionization current, one of the poles of the secondary coil of the spark coils is connected to the spark plugs in an ordinary manner. The other pole isn't grounded directly but connected to an 80 V voltage. This means that an 80 V voltage is fielded over the spark gap of the spark plugs, except when the spark is fired. When combustion has occurred the temperature in the combustion chamber is very high. The gases are formed as ions and start to conduct electrical current. This results in a current flowing in the spark plug gap (without resulting in a spark). The ionization current is measured in pair, cylinder one and two is one pair and cylinder three and four in the other pair. If combustion occurs in cylinder one or two the ignition cassette is sending a battery voltage (B+) pulse to the ECU, pin 17. If the combustion takes place in cylinder three or four the B+ pulse is feed to pin 18 in ECU. If the crankshaft position sensor is indicating that cylinders one and four is at TDC and a B+ pulse enters the ECU via pin 17 simultaneously, then the ECU know that it is cylinder one that has ignited. At start the ECU doesn't know which cylinder is in the compression stroke. Ignition is initiated in both cylinder one and four and 180° crank shaft degrees later sparks in cylinder two and three are fired. As soon as combustion signals enters the ECU via pin 17 and pin 18 the ignition and fuel injection is synchronized to the engines firing order. The combustion signals are also used to detect misfires.

Ignition maps

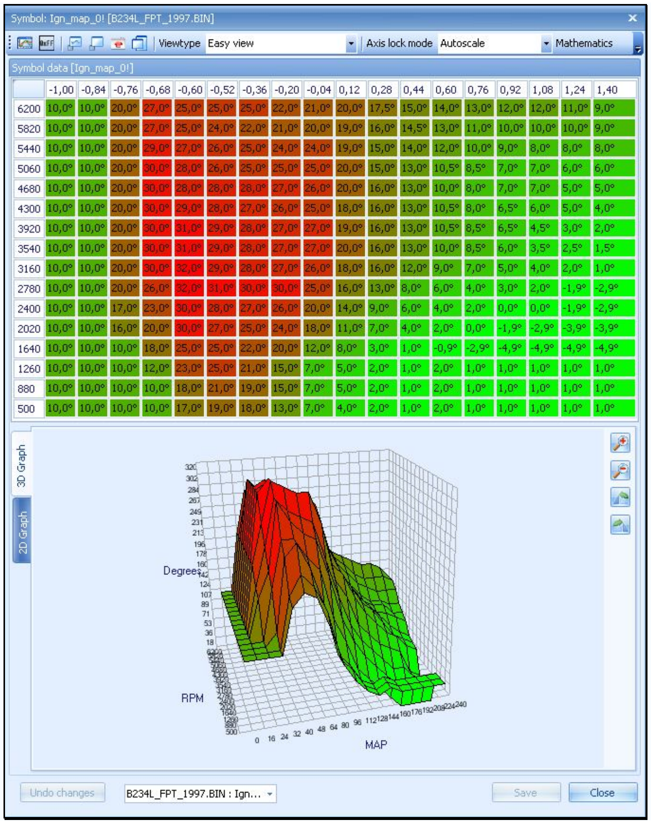
The ignition timing is vital to achieve MBT (M*aximum Brake Torque*) by using the minimum ignition spark advance. In theory this is simple: For each RPM and load point, advance the ignition timing until you reach the peak torque, the more ignition advance BTDC (*Before Top Dead Centre*) the more torque you will get until the resulting combustion pressure begins to slow down the upward moving piston and the torque starts to decrease.

The more ignition advance you use the more NOx emissions you will generate resulting from an increase in combustion chamber temperature, this is one factor to take into consideration when establishing the correct ignition advance. One other factor is the end gases, gases that are formed from deterioration of the fuel present in the combustion chamber from the compression heat. The end gases can auto ignite when the spark is fired from the spark plug causing two flame fronts in the combustion chamber, these flame fronts collide and form a knocking sound as the resulting flame front is oscillating in the pressure traverse. This could easily damage the engine internals as the peak pressure can be higher than the pressure occurring during MBT. The higher combustion chamber temperature from an increased ignition timing can lead to "hot spots" in the combustion chamber and thus ignite the Air/Fuel mixture separate from the spark plug and cause the unwanted knocking. This detonation can almost instantaneously kill your engine since the ignition may take place at a totally wrong time. A fuel with higher octane rating is more resistant to these engine damaging phenomena. This is reflected in the differences between the main ignition maps on B234R and B234L

Also bear in mind that since the emissions controls prevent the engine manufacturers to reach MBT they could save some money and weight on the pistons, thereby giving an ambitious tuner a lighter wallet as a result from broken pistons. The collapsed piston roof or cracked piston ring is a direct result from high combustion pressure, not necessarily from knocking, but only from the normal increase in pressure resulting from driving at MBT without knocking. This is not a known problem in engines with T5 though.

The Ethanol fuel E85 has different combustion characteristics resulting in more ignition timing advance at low RPM´s and less ignition advance at high RPM´s compared to gasoline. This is the result of the higher combustion rate of Ethanol and its less initial combustion rate. I.e. it's harder to ignite but when it ignites it burns faster than gasoline. This is the single most error E85 converters do, advancing the timing to restore the torque and not put the ignition advance later when needed, thereby putting unnecessary stress to the piston by passing MBT.

Different stroke requires different ignition advance since the different length of the piston rod changes the geometry of the piston, piston rod and crank shaft system. When the piston rod is shorter than infinite length the reciprocating motion doesn't describe the desired sine curve, rather it is a triangle wave with flat tops. The different stroke doesn't change the ignition advance in any easy predictable way. See the difference between the ignition advance on 2.0 LPT and 2.3 LPT engines.



Symbol: Ign_map_0! [B234L_FPT_1997.BIN]

Viewtype Easy view

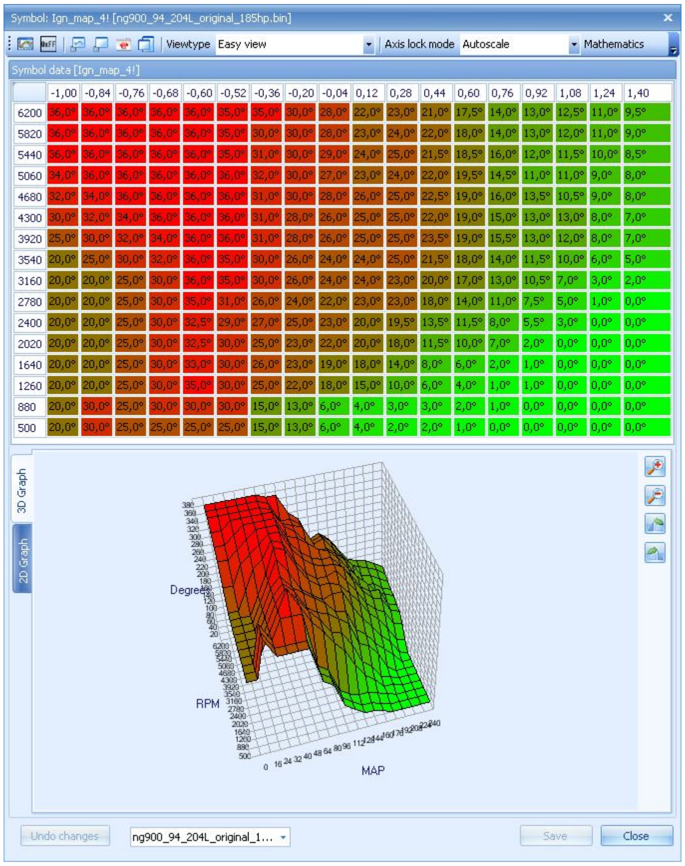
Axis lock mode Autoscale

Symbol data [Ign_map_0!]

	-1,00	-0,84	-0,76	-0,68	-0,60	-0,52	-0,36	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24	1,40
6200	10,0°	10,0°	20,0°	27,0°	25,0°	25,0°	25,0°	22,0°	21,0°	20,0°	17,5°	15,0°	14,0°	13,0°	12,0°	12,0°	11,0°	9,0°
5820	10,0°	10,0°	20,0°	27,0°	25,0°	24,0°	22,0°	21,0°	20,0°	19,0°	16,0°	14,5°	13,0°	11,0°	10,0°	10,0°	10,0°	9,0°
5440	10,0°	10,0°	20,0°	29,0°	27,0°	26,0°	25,0°	24,0°	24,0°	19,0°	15,0°	14,0°	12,0°	10,0°	9,0°	8,0°	8,0°	8,0°
5060	10,0°	10,0°	20,0°	30,0°	28,0°	26,0°	25,0°	25,0°	25,0°	20,0°	15,0°	13,0°	10,5°	8,5°	7,0°	7,0°	6,0°	6,0°
4680	10,0°	10,0°	20,0°	30,0°	28,0°	28,0°	28,0°	27,0°	26,0°	20,0°	16,0°	13,0°	10,0°	8,0°	7,0°	7,0°	5,0°	5,0°
4300	10,0°	10,0°	20,0°	30,0°	29,0°	28,0°	27,0°	26,0°	25,0°	18,0°	16,0°	13,0°	10,5°	8,0°	6,5°	6,0°	5,0°	4,0°
3920	10,0°	10,0°	20,0°	30,0°	31,0°	29,0°	28,0°	27,0°	27,0°	19,0°	16,0°	13,0°	10,5°	8,5°	6,5°	4,5°	3,0°	2,0°
3540	10,0°	10,0°	20,0°	30,0°	31,0°	29,0°	28,0°	27,0°	27,0°	20,0°	16,0°	13,0°	10,0°	8,5°	6,0°	3,5°	2,5°	1,5°
3160	10,0°	10,0°	20,0°	30,0°	32,0°	29,0°	28,0°	27,0°	26,0°	18,0°	16,0°	12,0°	9,0°	7,0°	5,0°	4,0°	2,0°	1,0°
2780	10,0°	10,0°	20,0°	26,0°	32,0°	31,0°	30,0°	30,0°	25,0°	16,0°	13,0°	8,0°	6,0°	4,0°	3,0°	2,0°	-1,9°	-2,9°
2400	10,0°	10,0°	17,0°	23,0°	30,0°	28,0°	27,0°	26,0°	20,0°	14,0°	9,0°	6,0°	4,0°	2,0°	0,0°	0,0°	-1,9°	-2,9°
2020	10,0°	10,0°	16,0°	20,0°	30,0°	27,0°	25,0°	24,0°	18,0°	11,0°	7,0°	4,0°	2,0°	0,0°	-1,9°	-2,9°	-3,9°	-3,9°
1640	10,0°	10,0°	10,0°	18,0°	25,0°	25,0°	22,0°	20,0°	12,0°	8,0°	3,0°	1,0°	-0,9°	-2,9°	-4,9°	-4,9°	-4,9°	-4,9°
1260	10,0°	10,0°	10,0°	12,0°	23,0°	25,0°	21,0°	15,0°	7,0°	5,0°	2,0°	1,0°	2,0°	1,0°	1,0°	1,0°	1,0°	1,0°
880	10,0°	10,0°	10,0°	10,0°	18,0°	21,0°	19,0°	15,0°	7,0°	5,0°	2,0°	1,0°	2,0°	1,0°	1,0°	1,0°	1,0°	1,0°
500	10,0°	10,0°	10,0°	10,0°	17,0°	19,0°	18,0°	13,0°	7,0°	4,0°	2,0°	1,0°	2,0°	1,0°	1,0°	1,0°	1,0°	1,0°

1: Normal ignition map (warm engine)

On the B204E in NG900 the ignition timing is well advanced in low RPM/Load during warm up to make the engine work harder and inefficient, more heat is generated inside the cylinders. This inefficiency enables the catalyst to heat up and start working quicker. The warm up ignition map is Ign_map_4! and the normal ignition map is Ign_map_0!



Symbol: Ign_map_4! [ng900_94_204L_original_185hp.bin]

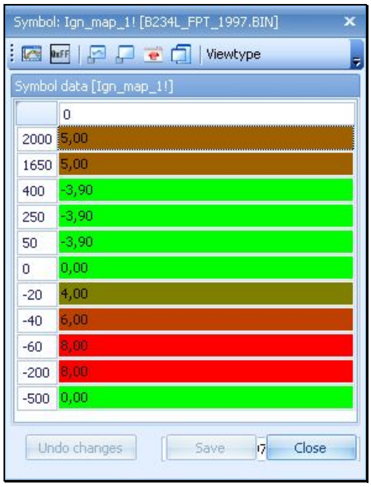
Viewtype Easy view | Axis lock mode Autoscale | Mathematics

Symbol data [Ign_map_4!]

	-1,00	-0,84	-0,76	-0,68	-0,60	-0,52	-0,36	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24	1,40
6200	36,0°	36,0°	36,0°	36,0°	36,0°	35,0°	35,0°	30,0°	28,0°	22,0°	23,0°	21,0°	17,5°	14,0°	13,0°	12,5°	11,0°	9,5°
5820	36,0°	36,0°	36,0°	36,0°	36,0°	35,0°	30,0°	30,0°	28,0°	23,0°	24,0°	22,0°	18,0°	14,0°	13,0°	12,0°	11,0°	9,0°
5440	36,0°	36,0°	36,0°	36,0°	36,0°	35,0°	31,0°	30,0°	29,0°	24,0°	25,0°	21,5°	18,5°	16,0°	12,0°	11,5°	10,0°	8,5°
5060	34,0°	36,0°	36,0°	36,0°	36,0°	36,0°	32,0°	30,0°	27,0°	23,0°	24,0°	22,0°	19,5°	14,5°	11,0°	11,0°	9,0°	8,0°
4680	32,0°	34,0°	36,0°	36,0°	36,0°	36,0°	31,0°	30,0°	28,0°	26,0°	25,0°	22,5°	19,0°	16,0°	13,5°	10,5°	9,0°	8,0°
4300	30,0°	32,0°	34,0°	36,0°	36,0°	36,0°	31,0°	28,0°	26,0°	25,0°	25,0°	22,0°	19,0°	15,0°	13,0°	13,0°	8,0°	7,0°
3920	25,0°	30,0°	32,0°	34,0°	36,0°	36,0°	31,0°	28,0°	26,0°	25,0°	25,0°	23,5°	19,0°	15,5°	13,0°	12,0°	8,0°	7,0°
3540	20,0°	25,0°	30,0°	32,0°	36,0°	35,0°	30,0°	26,0°	24,0°	24,0°	25,0°	21,5°	18,0°	14,0°	11,5°	10,0°	6,0°	5,0°
3160	20,0°	20,0°	25,0°	30,0°	36,0°	35,0°	30,0°	26,0°	24,0°	24,0°	23,0°	20,0°	17,0°	13,0°	10,5°	7,0°	3,0°	2,0°
2780	20,0°	20,0°	25,0°	30,0°	35,0°	31,0°	26,0°	24,0°	22,0°	23,0°	23,0°	18,0°	14,0°	11,0°	7,5°	5,0°	1,0°	0,0°
2400	20,0°	20,0°	25,0°	30,0°	32,5°	29,0°	27,0°	25,0°	23,0°	20,0°	19,5°	13,5°	11,5°	8,0°	5,5°	3,0°	0,0°	0,0°
2020	20,0°	20,0°	25,0°	30,0°	32,5°	30,0°	25,0°	23,0°	22,0°	20,0°	18,0°	11,5°	10,0°	7,0°	2,0°	0,0°	0,0°	0,0°
1640	20,0°	20,0°	25,0°	30,0°	33,0°	30,0°	26,0°	23,0°	19,0°	18,0°	14,0°	8,0°	6,0°	2,0°	1,0°	0,0°	0,0°	0,0°
1260	20,0°	20,0°	25,0°	30,0°	35,0°	30,0°	25,0°	22,0°	18,0°	15,0°	10,0°	6,0°	4,0°	1,0°	1,0°	0,0°	0,0°	0,0°
880	20,0°	30,0°	25,0°	30,0°	30,0°	30,0°	15,0°	13,0°	6,0°	4,0°	3,0°	3,0°	2,0°	1,0°	0,0°	0,0°	0,0°	0,0°
500	20,0°	30,0°	25,0°	25,0°	25,0°	25,0°	15,0°	13,0°	6,0°	4,0°	2,0°	2,0°	1,0°	0,0°	0,0°	0,0°	0,0°	0,0°

2: Warmup ignition map

When idling the idle ignition map "Ign_map_1!" is used to control the engine´s torque at idle, this compensates for rough idling.



Symbol: Ign_map_1! [B234L_FPT_1997.BIN]

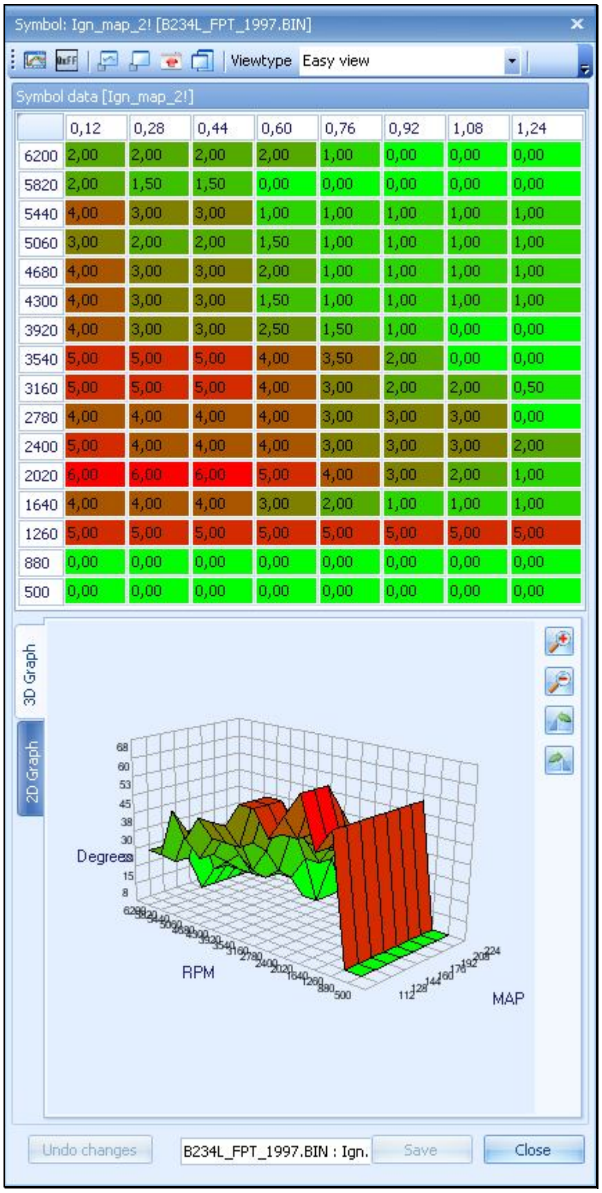
Viewtype

Symbol data [Ign_map_1!]

	0
2000	5,00
1650	5,00
400	-3,90
250	-3,90
50	-3,90
0	0,00
-20	4,00
-40	6,00
-60	8,00
-200	8,00
-500	0,00

3: Idle ignition map

In case of detonation/knocking the ignition timing is retarded by 1,5° in steps for that specific cylinder until the knocking stops. This is done to a maximum retardation of 12°. In an event of all four cylinders knocking and it´s the same ignition timing retardation, more fuel is injected and ignition timing is restored. If manifold pressure (MAP) is over 140 kPa when knocking occurs the ignition map is switched to Ign_map_2! This map indicates how much ignition timing to pull (retard) from the original values as a countermeasure to stop the knocking.



Symbol: Ign_map_2! [B234L_FPT_1997.BIN]

Viewtype: Easy view

Symbol data [Ign_map_2!]

	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24
6200	2,00	2,00	2,00	2,00	1,00	0,00	0,00	0,00
5820	2,00	1,50	1,50	0,00	0,00	0,00	0,00	0,00
5440	4,00	3,00	3,00	1,00	1,00	1,00	1,00	1,00
5060	3,00	2,00	2,00	1,50	1,00	1,00	1,00	1,00
4680	4,00	3,00	3,00	2,00	1,00	1,00	1,00	1,00
4300	4,00	3,00	3,00	1,50	1,00	1,00	1,00	1,00
3920	4,00	3,00	3,00	2,50	1,50	1,00	0,00	0,00
3540	5,00	5,00	5,00	4,00	3,50	2,00	0,00	0,00
3160	5,00	5,00	5,00	4,00	3,00	2,00	2,00	0,50
2780	4,00	4,00	4,00	4,00	3,00	3,00	3,00	0,00
2400	5,00	4,00	4,00	4,00	3,00	3,00	3,00	2,00
2020	6,00	6,00	6,00	5,00	4,00	3,00	2,00	1,00
1640	4,00	4,00	4,00	3,00	2,00	1,00	1,00	1,00
1260	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00
880	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
500	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

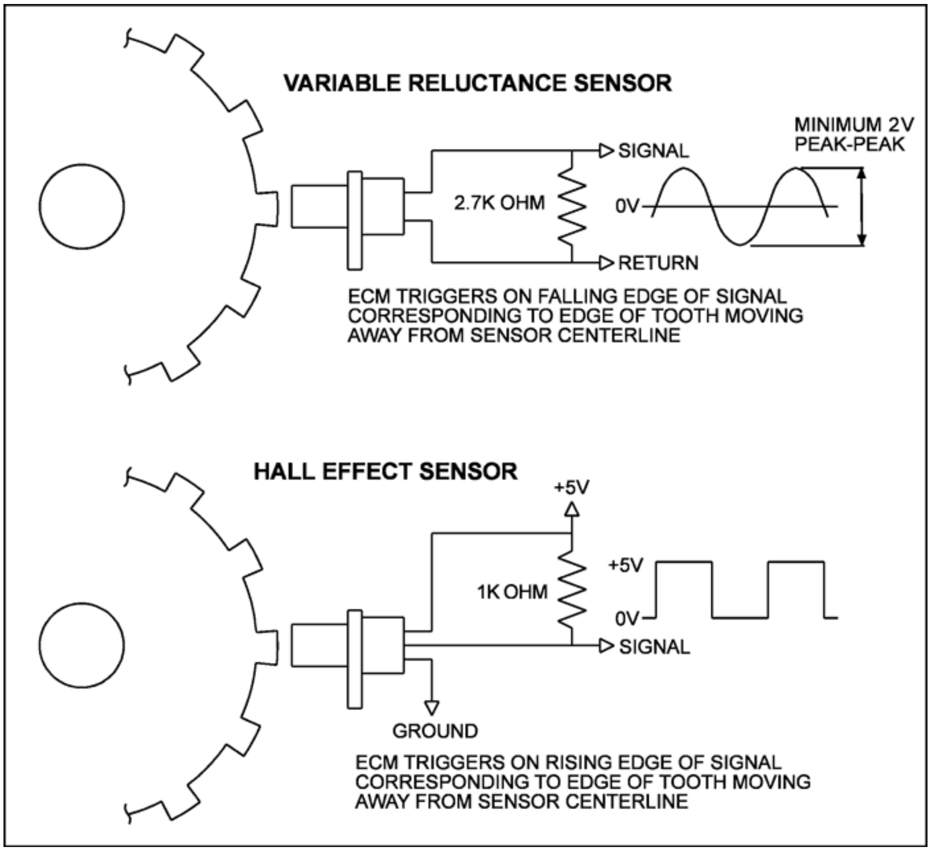
Altering ignition maps

To be able to tune these maps you must firstly use common sense. Then you must install an EGT meter. The Exhaust Gas Temperature is affected on the ignition timing. The further away from TDC the spark fires, the cooler the exhaust gases will be but as described earlier in the chapter the risk of detonation increases the more you advance the ignition timing. At high charge pressure you must retard the ignition to avoid detonation, this results in an increase in EGT. For a SAAB T5 engine the target EGT starts at 930° F (~500° C) and stops at 1795° F (~980° C). To lower this temperature the ignition can be advanced, as little as 0,5° might be enough. Adding more fuel at this RPM/Load point is also lowering the EGT. In addition you might consider using water injection or change fuel to a more fast burning fuel e.g. Ethanol or E85. To be totally safe when tuning you use a cylinder pressure transducer to find out your maximum pressure but this might be over your budget as a reliable kit costs as much as a few engines.

Crankshaft position sensor

⚠ Trionic T5.2 uses a different crankshaft position sensor than Trionic T5.5.
The early (T5.2) models use a HALL effect sensor while the later models (T5.5) uses a variable reluctance sensor.

The HALL effect sensor produces a square wave output while the VR type produces a sine wave output as shown in the picture below. The VR type output voltage is converted by T5.5 by using the LM1815. This results in a similar output as the HALL effect sensor. T5 has 62 pulses per engine rotation from which 2 teeth are missing at 117° BTDC on cylinder 1 to determine TDC.


VARIABLE RELUCTANCE SENSOR
MINIMUM 2V
PEAK-PEAK
SIGNAL
2.7K OHM
0V
RETURN
ECM TRIGGERS ON FALLING EDGE OF SIGNAL
CORRESPONDING TO EDGE OF TOOTH MOVING
AWAY FROM SENSOR CENTERLINE
HALL EFFECT SENSOR
+5V
+5V
1K OHM
0V
SIGNAL
GROUND
ECM TRIGGERS ON RISING EDGE OF SIGNAL
CORRESPONDING TO EDGE OF TOOTH MOVING
AWAY FROM SENSOR CENTERLINE

HOWTO: Get your ignition advance under control

After mapping fuel you can start working at optimizing ignition timing. To get the most of the engine in a safe way you need – of course – at least a wideband lambda controller and an EGT gauge. The latter is firstly to check if you're not passing 950° C and secondly to be able to see the effect your ignition trimming has on EGT values. I find that the easiest way to map ignition timing is to get fuel mapped properly first, then boost control to a very stable level and then request boost to match on of the ignition map columns (for example 1.2 bar). Do a run and see if knock occurs using T5Suite or the dashboard. If you have no knock, increase timing a little bit and rerun. Continue to map all relevant cells until you are at approximately 1 degree timing from knock (so, if knock occurs at 7 degrees, you should map for 6 degrees advance).
Start at a low advance (known to work well) and work your way up from there, not the other way around! You should not only feel the increase in power this way but also see EGT drop a bit in the process which is good. If you run the car on RON95 normally you should map for RON95. If you fuel up on RON98 always you can map for RON98 of course. Use common sense in this!

Tips: Knock can be seen in either the log files or in the knock_count_map and knock_count_cylx variables. If you have knock in one specific cylinder only (or the majority at least) deal with that first before proceeding on the main ignition map. One of the injectors may be off or you might have a fouled up spark plug. Swap them one by one and see if the knocking travels to the other cylinder as well.

Boost

Basic charging pressure

Basic charging pressure is fundamental for charging pressure control. Basic charging pressure is mechanically adjusted on the actuators pushrod between the actuator and the waste gate. At too low basic charging pressure the engine doesn't rev up as expected when the throttle is opened quickly. At to high basic charging pressure a negative adaption occurs and maximum charging pressure cannot be achieved. In addition there is a substantial risk of engine damage since the charging pressure can't be lowered enough when regulating with attention to pre ignition/pinging. Basic charging pressure shall be 0,40 ±0,03 bar (5,80 ±0,43 PSI). After adjustment the push rod must have at least three turns (~3 mm) pre tension when connecting to the waste gate lever. The purpose with that is to make sure that the waste gate is held close when not affected. On new turbo chargers the basic charging pressure tends to be near or spot on the upper tolerance when the pre tension is two turns. The pre tension may never be less than three turns (~3 mm). When checking the basic charging pressure it shall be noted that the pressure decreases at high RPM and increases at low outside temperatures.

Charging pressure regulation

Charging pressure regulation utilises a two coiled three way solenoid valve pneumatically connected with hoses to the turbo chargers waste gate, the turbo chargers outlet and the compressor's inlet. The solenoid valve is electrically supplied from +54 via fuse 13 and is controlled by the ECU via its pin 26 and pin 2. These pins are driven by MOSFETs. The control voltage is pulse width modulated (PWM) at 90 Hz below 2500 RPM and 70 Hz above 2500 RPM. The rationale for this change is to avoid resonance phenomena in the pneumatic hoses. By grounding pin 2 longer than pin 26 the charging pressure is decreased and vice verse, when pin 26 is grounded longer than pin 2 the charging pressure is increased. To be able to regulate the charging pressure the ECU must at first calculate a requested pressure, a pressure value that the system must strive for. This is done by taking a pre programmed value (matrix of values established in respect of RPM and throttle opening). When one or both of the following criteria are met, a limitation of the charging pressure is set.

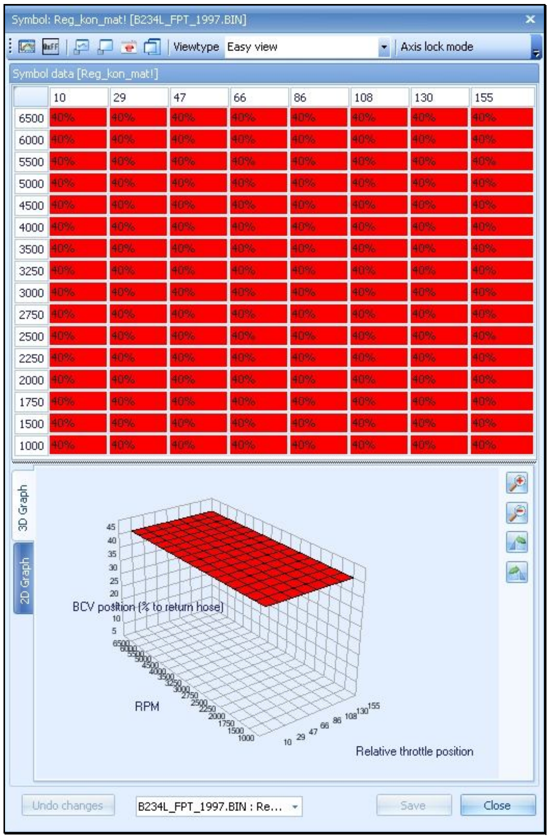
- In first, second and reverse gear there is a RPM dependable maximum value. The ECU calculates which gear that is in use by comparing the speed of the automobile and the engines RPM.
- When pre ignition/pinging occurs a maximum charge pressure is set on the basis of a mean value from each cylinders retarding of the ignition.

One or both of the following criteria initiates a lowering of the charging boost pressure to basic boost pressure.

- When the brake pedal is pressed down and pin 15 (brake light switch) on the ECU is supplied with battery voltage.
- Certain fault codes are set (Faulty throttle position sensor (TPS), pressure sensor, pre ignition/pinging signal or charging pressure regulation) or low battery voltage.

To be able to electrically regulate the charging pressure we must decide the relationship between the air that is sent to the wastegate as the working media to actuate the wastegate valve. When driving under ideal conditions this relationship is enough to get a stable charge pressure. This map is of course not necessary on LPT´s since the wastegate is not electrically/pneumatically actuated. This map also controls how fast the turbo can reach the desired pressure with some respect to the durability of the transmission. Tuning this map is necessary when changing turbo charger or its working conditions such as: 3" down pipe, porting the cylinder head or changing intercooler. The aim when tuning this map is to get a stable pressure build up without any overshoot when reaching the desired pressure. A calibrated charge pressure meter is a must when undertaking this task.

The regulation table determines the bias (sort of pre-tension) for the pulse width signal to the boost control valve. It can be considered an offset for the PID controller. So PID and regulation bias determine the pulse width on the boost control valve taking the pressure error (boost request minus actual boost) as an input value.


Symbol: Reg_kon_mat! [B234L_FPT_1997.BIN]

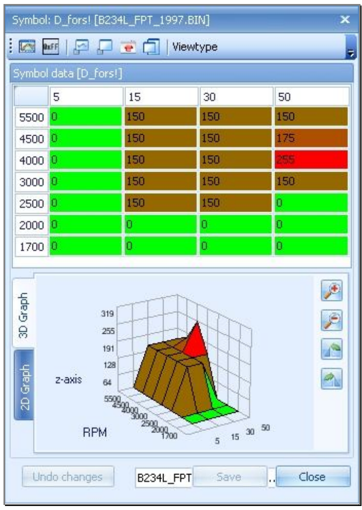
Viewtype: Easy view

Symbol data [Reg_kon_mat!]

	10	29	47	66	86	108	130	155
6500	40%	40%	40%	40%	40%	40%	40%	40%
6000	40%	40%	40%	40%	40%	40%	40%	40%
5500	40%	40%	40%	40%	40%	40%	40%	40%
5000	40%	40%	40%	40%	40%	40%	40%	40%
4500	40%	40%	40%	40%	40%	40%	40%	40%
4000	40%	40%	40%	40%	40%	40%	40%	40%
3500	40%	40%	40%	40%	40%	40%	40%	40%
3250	40%	40%	40%	40%	40%	40%	40%	40%
3000	40%	40%	40%	40%	40%	40%	40%	40%
2750	40%	40%	40%	40%	40%	40%	40%	40%
2500	40%	40%	40%	40%	40%	40%	40%	40%
2250	40%	40%	40%	40%	40%	40%	40%	40%
2000	40%	40%	40%	40%	40%	40%	40%	40%
1750	40%	40%	40%	40%	40%	40%	40%	40%
1500	40%	40%	40%	40%	40%	40%	40%	40%
1000	40%	40%	40%	40%	40%	40%	40%	40%

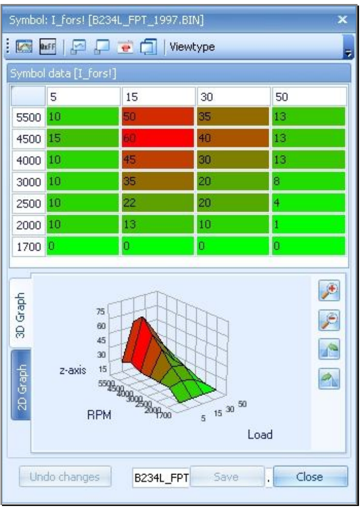
Axis lock mode
3D Graph
2D Graph
BCV position (% to return hose)
45
40
35
30
25
20
10
5
RPM
Relative throttle position
Undo changes
B234L_FPT_1997.BIN : Re...
Save
Close

Image 8: Boost regulation table


Symbol: D_fors! [B234L_FPT_1997.BIN]

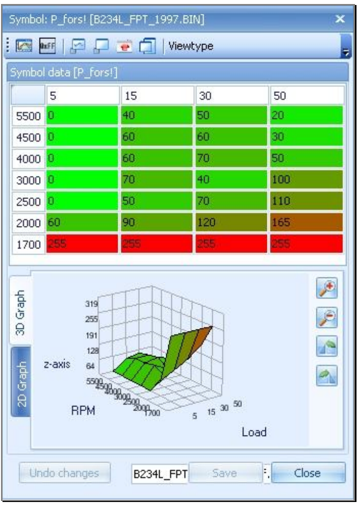
Symbol data [D_fors!]

	5	15	30	50
5500	0	150	150	150
4500	0	150	150	175
4000	0	150	150	255
3000	0	150	150	150
2500	0	150	150	0
2000	0	0	0	0
1700	0	0	0	0


Symbol: I_fors! [B234L_FPT_1997.BIN]

Symbol data [I_fors!]

	5	15	30	50
5500	10	50	35	13
4500	15	60	40	13
4000	10	45	30	13
3000	10	35	20	8
2500	10	22	20	4
2000	10	13	10	1
1700	0	0	0	0


Symbol: P_fors! [B234L_FPT_1997.BIN]

Symbol data [P_fors!]

	5	15	30	50
5500	0	40	50	20
4500	0	60	60	30
4000	0	60	70	50
3000	0	70	40	100
2500	0	50	70	110
2000	60	90	120	165
1700	255	255	255	255

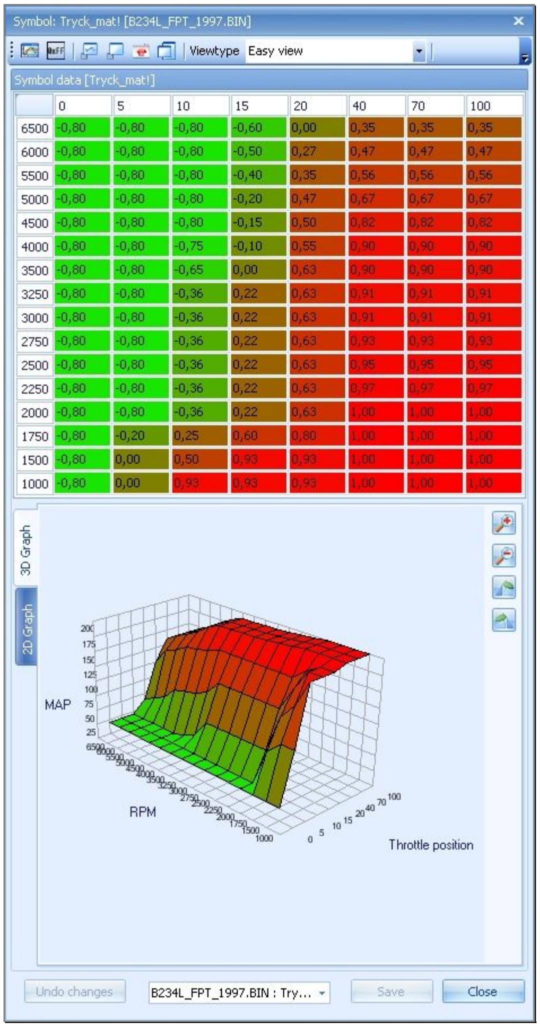
Above you see an example of the PID control tables that are used in a stock firmware.

Computing adaption

When the required charging pressure has finally been calculated it is converted to the PWM signal that controls the solenoid valve. The ECU then controls that the actual pressure (measured by the pressure sensor) corresponds with the required pressure. If needed the PWM is fine tuned by multiplication of a correction factor. The correction factor (adaption) is then stored in the memory of the ECU and is always used in the calculation of the PWM signal. The rationale with this is to make sure that the actual pressure will be equal to the required pressure as soon as possible after a change of the load has occurred.

Boost request

Trionic tries to reach the pressure values for a given RPM and throttle position that is contained in the map "Tryck_mat" (or "Tryck_mat_a" for automatic transmissions).
The stock boost request map for a B234L (200bhp, Garrett T25 turbo) is shown in *image 8*.
You can see that the requested pressure drops when RPM values go over ~4500. This is because the little T25 turbo cannot maintain higher pressure values at these engine speeds, it is simply too small for that. In modified cars containing a larger turbocharger these values will keep rising with engine speed.



Symbol: Tryck_mat! [B234L_FPT_1997.BIN]

Viewtype Easy view

Symbol data [Tryck_mat!]

	0	5	10	15	20	40	70	100
6500	-0,80	-0,80	-0,80	-0,60	0,00	0,35	0,35	0,35
6000	-0,80	-0,80	-0,80	-0,50	0,27	0,47	0,47	0,47
5500	-0,80	-0,80	-0,80	-0,40	0,35	0,56	0,56	0,56
5000	-0,80	-0,80	-0,80	-0,20	0,47	0,67	0,67	0,67
4500	-0,80	-0,80	-0,80	-0,15	0,50	0,82	0,82	0,82
4000	-0,80	-0,80	-0,75	-0,10	0,55	0,90	0,90	0,90
3500	-0,80	-0,80	-0,65	0,00	0,63	0,90	0,90	0,90
3250	-0,80	-0,80	-0,36	0,22	0,63	0,91	0,91	0,91
3000	-0,80	-0,80	-0,36	0,22	0,63	0,91	0,91	0,91
2750	-0,80	-0,80	-0,36	0,22	0,63	0,93	0,93	0,93
2500	-0,80	-0,80	-0,36	0,22	0,63	0,95	0,95	0,95
2250	-0,80	-0,80	-0,36	0,22	0,63	0,97	0,97	0,97
2000	-0,80	-0,80	-0,36	0,22	0,63	1,00	1,00	1,00
1750	-0,80	-0,20	0,25	0,60	0,80	1,00	1,00	1,00
1500	-0,80	0,00	0,50	0,93	0,93	1,00	1,00	1,00
1000	-0,80	0,00	0,93	0,93	0,93	1,00	1,00	1,00

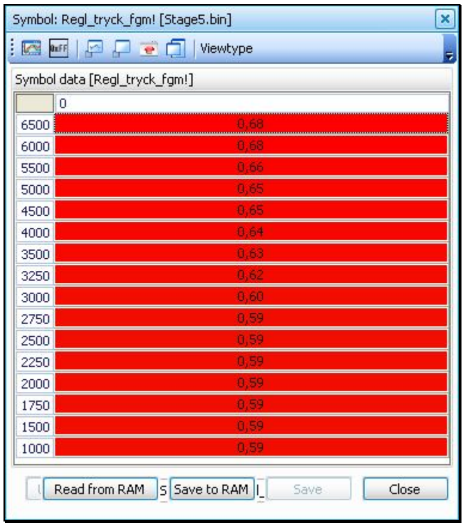
B234L_FPT_1997.BIN : Try...

Image 9: Boost request map for B234L

Limiters

To prevent damage to the gearboxes there is a charge pressure limiter on the reverse, first and second gear on manual gearboxes. The limiter on automatic transmission is active on the first and reverse stage. This reduces the engines torque as while the downshift of the RPM on the lower gears is increasing the torque so that we are able to get the car rolling without damaging any transmission details. Main boost request comes from Tryck_mat. If the boost request is higher than the value from the limiter, the limiter value will become the actual boost request.

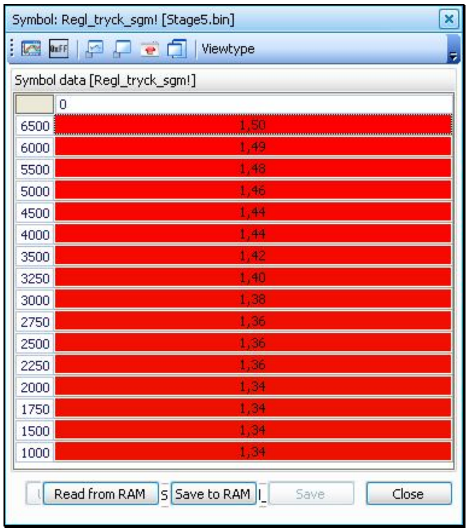
The first and second gear limiters are only present to prevent the ECU to use the main boost request map in first and second gear. If these limiters are set ABOVE the values in the main request map, the limiters are switched off.



Symbol: Regl_tryck_fgm! [Stage5.bin]

Symbol data [Regl_tryck_fgm!]

	0
6500	0,68
6000	0,68
5500	0,66
5000	0,65
4500	0,65
4000	0,64
3500	0,63
3250	0,62
3000	0,60
2750	0,59
2500	0,59
2250	0,59
2000	0,59
1750	0,59
1500	0,59
1000	0,59



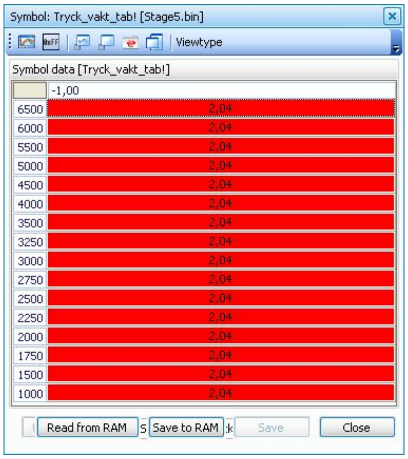
Symbol: Regl_tryck_sgm! [Stage5.bin]

Symbol data [Regl_tryck_sgm!]

	0
6500	1,50
6000	1,49
5500	1,48
5000	1,46
4500	1,44
4000	1,44
3500	1,42
3250	1,40
3000	1,38
2750	1,36
2500	1,36
2250	1,36
2000	1,34
1750	1,34
1500	1,34
1000	1,34

Fuel cut limiter

To prevent damage to the engine caused by faulty hardware or overshooting boost, fuel will be cut off when charging pressure gets higher than set in the fuel cut map (Tryck_vakt_tab).



Symbol data [Tryck_vakt_tab!]

	-1,00
6500	2,04
6000	2,04
5500	2,04
5000	2,04
4500	2,04
4000	2,04
3500	2,04
3250	2,04
3000	2,04
2750	2,04
2500	2,04
2250	2,04
2000	2,04
1750	2,04
1500	2,04
1000	2,04

Image 10: Stock max. boost pressure table

This example shows that the maximum boost level is always 0xEB, this is the standard cut-off value of 1.35 bar.
0xFE would be highest possible, around 1.54 bar.
0xFF would turn off the fuel cut totally.

HOWTO: Get your boost under control

Boost control in Trionic 5 is quite simple on the one side and very complex on the other. If you are trying to regulate boost with a turbocharger that is closely related to the stock turbocharger (either T25 or TD04) it is quite simple to alter boost settings by altering the boost request map (Tryck_mat) if you stay within the limits for the turbo.

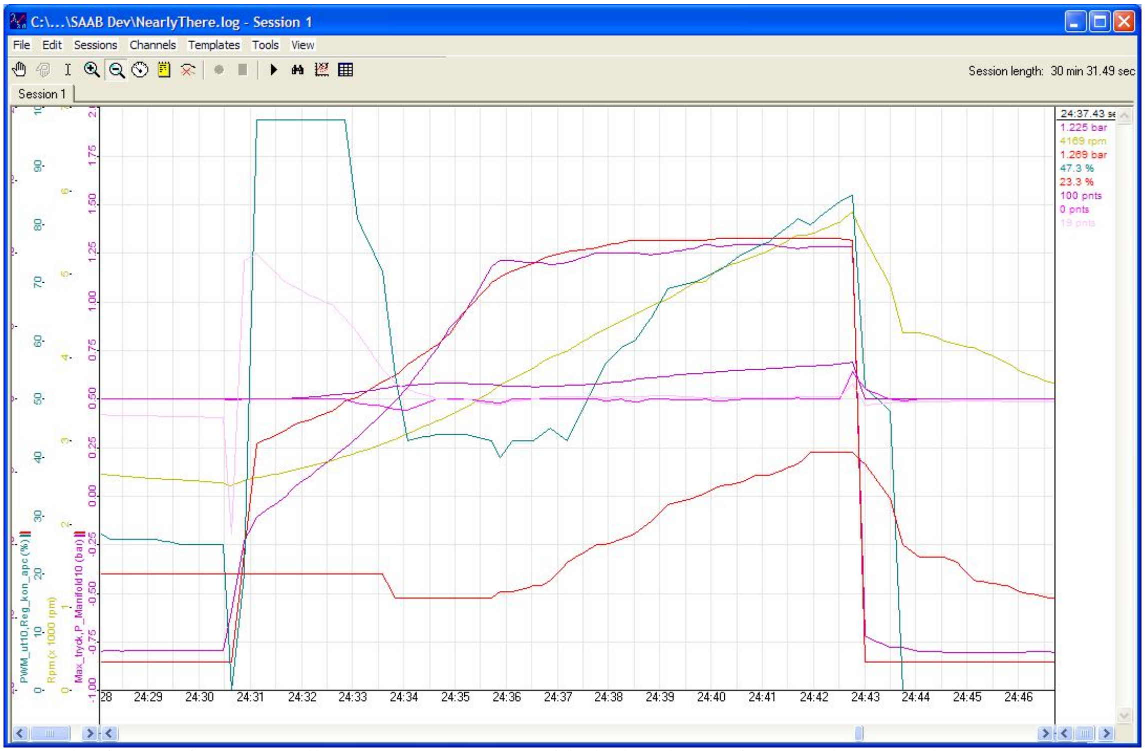
If you are using a larger than stock turbo (e.g. GT28, TD04-19T, GT3071R, Holset HX35/40w etc) then you might run into trouble getting your boost under control. There are several known issues that you can encounter here.

- Not getting your requested boost
- Overshooting when running @WOT from low RPM's
- Overshooting when running @WOT from high(er) RPM's

Not getting your requested boost

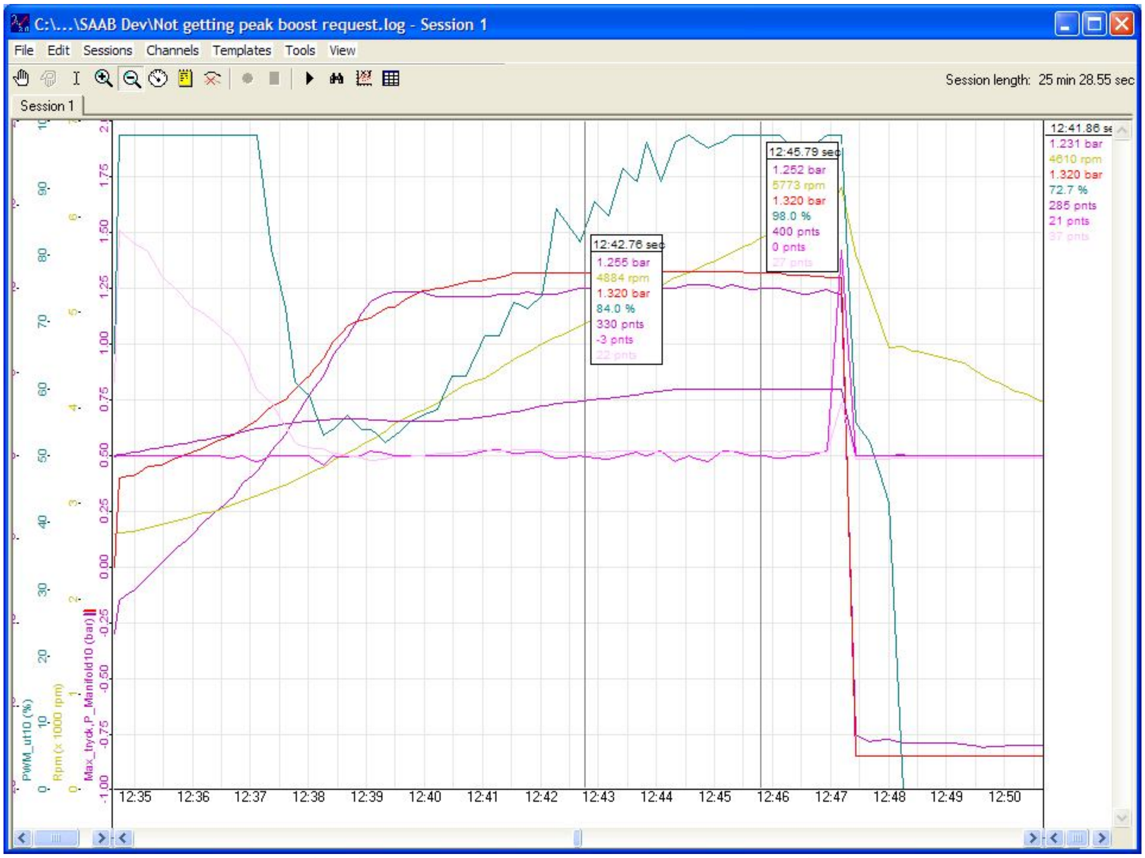
If you are not getting the amount of boost that you are requesting in Tryck_mat you might be looking at either a hardware problem or a software problem. To identify which kind you're having you will need to log information from the ECU through the canbus interface.
Software problems are a fact when "PWM_ut10" is NOT at its maximum value (98%) while you are not getting your request and you are in an RPM range where the turbo can definitely deliver more boost (test with W-hose disconnected).
If you decide it is a software failure you can start logging P_fak, I_fak, D_fak, Reg_kon_apc, PWM_ut10, Rpm, P_Manifold10, Ign_angle and Apc_decrese. If you see from the logging that Apc_decrese > 0 in the area where you're not getting the requested boost, you're having knock detection which is lowering boost. You should also see Ign_angle dropping a bit just before that point.

If Apc_decrese stays 0 and you're not getting your requested target AND PWM_ut10 < 98 then you should be able to see that I_fak is pretty low (<400 anyway). This could mean that you have to increase either reg_kon_mat values or I_fors values.



Example of a too soft PID controller. PWM_ut10 < 98%, I_fak climbing but not fast enough. Raise either Reg_kon_mat values or I_fors values.

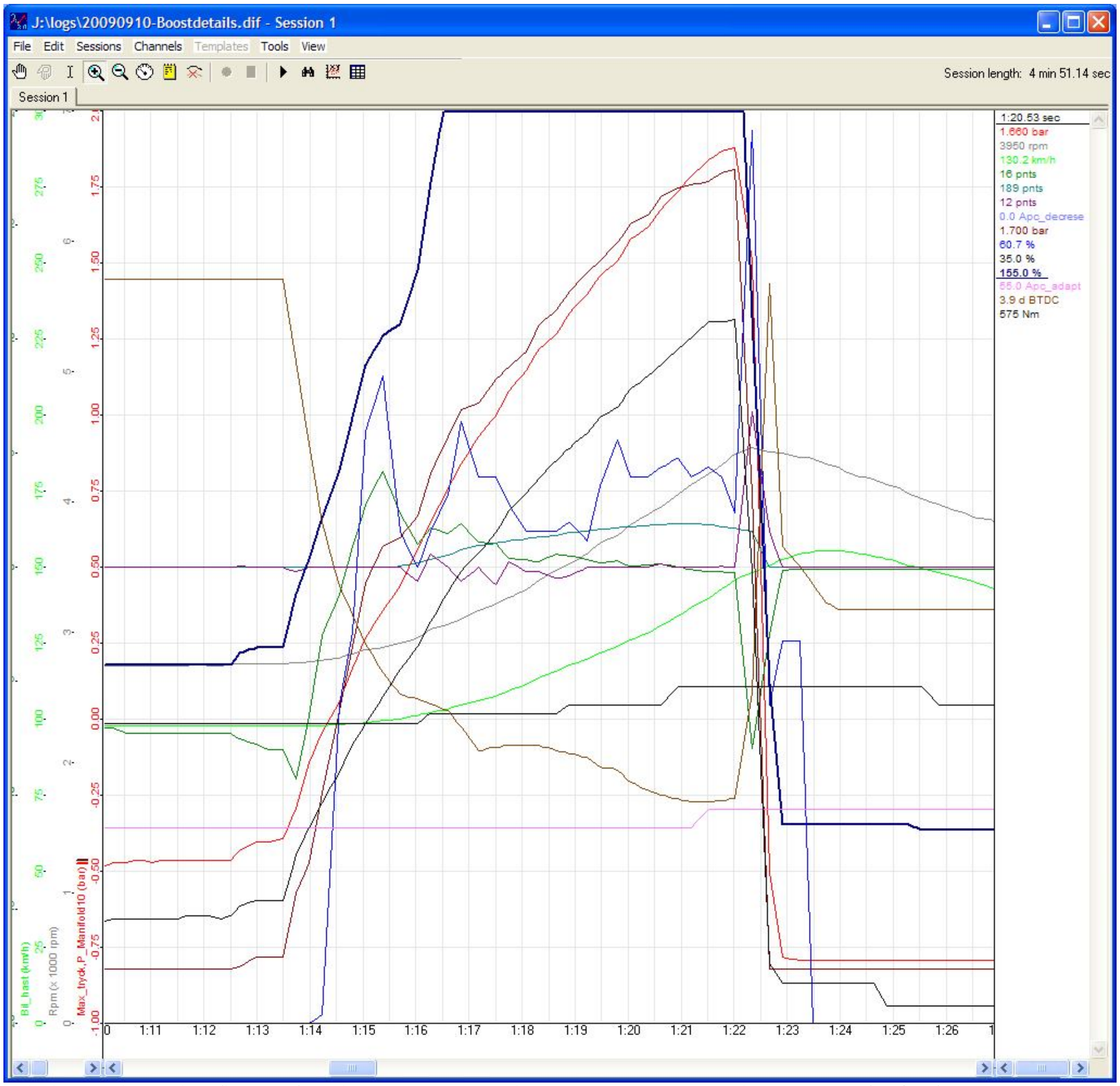
If you – on the other hand – see that PWM_ut10 is at its maximum and you're not getting the amount of boost you're requesting that means that the software it trying its best to meet you request and the problem should be searched for in the hardware. It might be vacuum leaks, a wastegate actuator that is too weak, to much backpressure, too large C-port in the APC valve, defective APC valve and then some other options. Searching for hardware failures is outside of the scope of this FAQ though.



Not reaching boost request even though PWM_ut10 is approx 98%. Hardware issues at hand.

Overshooting when running @WOT from low RPM's

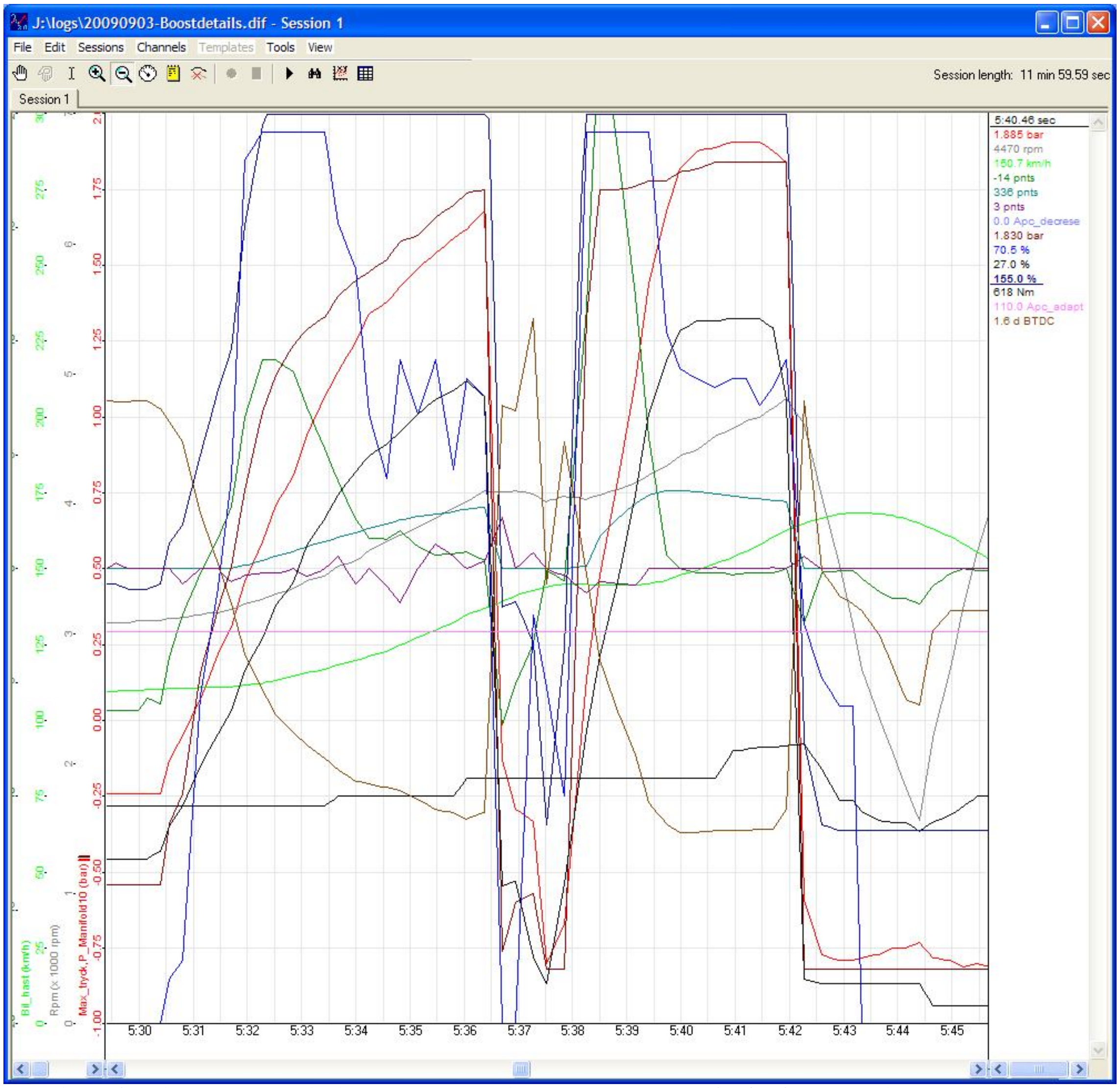
If you floor it from 2500 rpm and spool up is starting at 3000 rpm for example you might see a spike in the boost level just when you pass the point where you hit the requested boost. Let's assume that you request 1.8 bar and @ 4200 rpm you reach this level but boost overshoots to 1.9 bar @ 4250 rpm. This indicates that the spool characteristics defined in the PID maps are too aggressive. PID control should be smoother when almost reaching the requested boost. You can almost certainly map this correctly by decreasing I_fors values in the area below 4200 rpm.



A little overshoot when flooring from low RPM range. Solution is in trimming reg_kon_mat or I_fors/D_fors

Overshooting when running @WOT from high(er) RPM's

If you floor it from 4500 rpm and the turbo can deliver your requested boost, you might see a spike in the boost level just when you pass the point where you hit the requested boost. Let's assume that you request 1.8 bar and @ 4700 rpm you reach this level but boost overshoots to 1.9 bar @ 4750 rpm. This indicates that the spool characteristics defined in the PID maps are too aggressive. PID control should be smoother when almost reaching the requested boost. You can almost certainly map this correctly by decreasing P_fors and/or I_fors values in the area between 4500 and 5000 rpm.

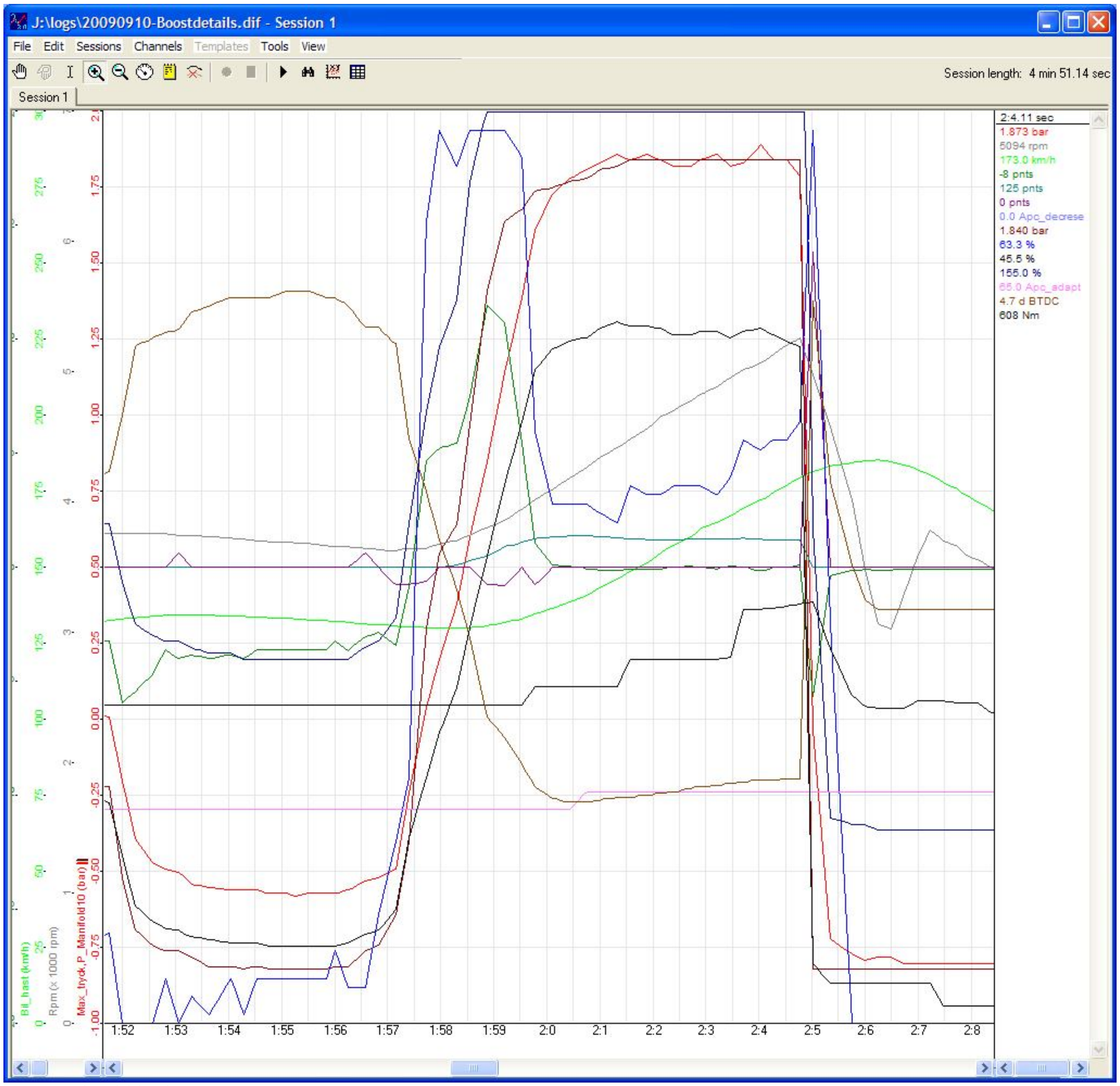


J:\logs\20090903-Boostdetails.dif - Session 1

Session length: 11 min 59.59 sec

5:40.46 sec
1.885 bar
4470 rpm
150.7 km/h
-14 pnts
336 pnts
3 pnts
0.0 Apc_decrese
1.830 bar
70.5 %
27.0 %
155.0 %
618 Nm
110.0 Apc_adapt
1.6 d BTDC

Overshoot after changing gear in higher RPM range. Solution is in trimming I_fors/D_fors or reg_kon_mat



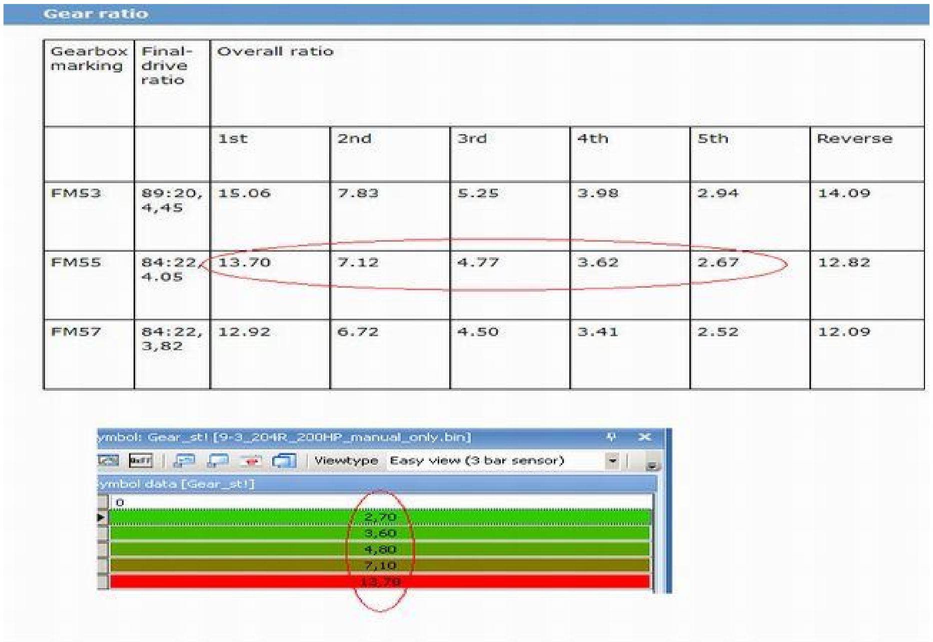
How it should look

Things to watch out for

I_fak should not equal I_fak_max (400 by default). If it does, you will have to raise reg_kon_mat in this area.
Apc_adapt should not equal 250 (its maximum value). If it does, adaption limits have been reached and something is not right in the PID mapping.

Gear ratio

Trionic calculates in which gear it is currently driving by using the engine speed, wheel pulses and the number of pulses per wheel revolution. Combined with the gear ratio settings (maps) in can make a calculation on which gear is currently engaged. This is important for boost limiting in 1st and 2nd gear.



Gear ratio

Gearbox marking	Final-drive ratio	Overall ratio					
		1st	2nd	3rd	4th	5th	Reverse
FM53	89:20, 4,45	15.06	7.83	5.25	3.98	2.94	14.09
FM55	84:22, 4.05	13.70	7.12	4.77	3.62	2.67	12.82
FM57	84:22, 3,82	12.92	6.72	4.50	3.41	2.52	12.09

ymbol: Gear_st! [9-3_204R_200HP_manual_only.bin]

Viewtype Easy view (3 bar sensor)

ymbol data [Gear_st!]

0
2,70
3,60
4,80
7,10
13,70

Tables Gear_st and Gear_ratio hold information about the different values for this calculation. Gear_st is the overall ratio per gear. In the example above the ratio for first gear is 13.7 according to the service manual for this car and model. The bottom value show in Gear_st is this exact value. 5th gear shows 2.67 in the service manual and so does the upper value in the table (2.70).
So, what does this figure mean? It is actually the ratio between engine revolutions and wheel revolutions. If the engine is running in 5th gear and with 3000 rpm with a gear ratio of 2.7 you will have 3000 / 2.7 wheel revolutions PER MINUTE. Given a wheel radius of e.g. 300 mm you will travel at a speed:

Wheel revolutions per minute = 3000/2.7 = 1111 rev/m
Wheel revolutions per second = 1111/60 = 18.5 wheel revs per second.
The distance traveled for a complete wheel rev is: 2 * pi * 300 = 1884 mm = 1.884 meter
The distance traveled per second will then be: 1.884 * 18.5 = 34.8 meter
The speed will be: 125 km/h

Variables

Program mode

This next table gives an overview of "program mode" options for Trionic. These are stored bitwise (one single bit switches functionality on or off) in the variable "Pgm_mod" and can be accessed by using the firmware options screen in T5Suite.

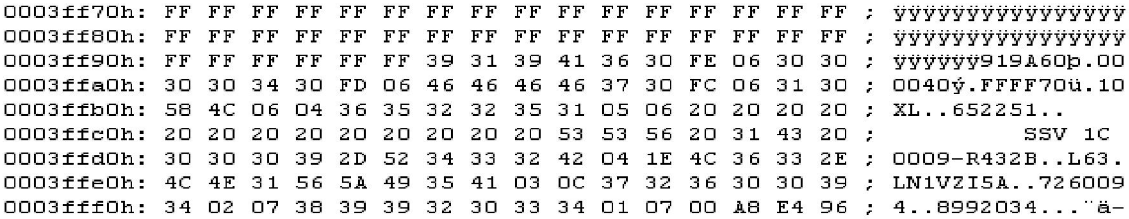
Byte	Bit	Mask	Description
0	0	0x01	Afterstart enrichment
0	1	0x02	WOT enrichment
0	2	0x04	Interpolation of delay map
0	3	0x08	Temperature compensation
0	4	0x10	Lambda control
0	5	0x20	Adaptivity
0	6	0x40	Idle control
0	7	0x80	Enrichment during start
1	0	0x01	Constant injection time (E51)
1	1	0x02	Lambda control during transients
1	2	0x04	Fuelcut
1	3	0x08	Constant injection time during idle
1	4	0x10	Accelerations enrichment
1	5	0x20	Decelerations enleanment
1	6	0x40	Car104
1	7	0x80	Adaptivity with closed throttle
2	0	0x01	Factor to lambda when throttle opening
2	1	0x02	Uses separate injection map during idle
2	2	0x04	Factor to lambda when AC is engaged
2	3	0x08	Throttle Acceleration Deceleration adjust simult MY95
2	4	0x10	Fuel adjusting during idle
2	5	0x20	Purge control enabled
2	6	0x40	Adaption of idle control
2	7	0x80	Lambda control during idle
3	0	0x01	Heated plates present
3	1	0x02	Automatic transmission
3	2	0x04	Load control
3	3	0x08	ETS
3	4	0x10	APC control
3	5	0x20	Higher idle during start
3	6	0x40	Global adaption enabled
3	7	0x80	Temperature compensation with active lambda control
4	0	0x01	Load buffer during idle
4	1	0x02	Constant idle ignition angle during gear one and two
4	2	0x04	No fuelcut in reverse, first and second gear
4	3	0x08	Airpump control
4	4	0x10	Normal asperated engine (non turbo)
4	5	0x20	Knock regulating disabled
4	6	0x40	Constant ignition angle
4	7	0x80	Purge valve MY94
5	7	0x80	VSS enabled
5	4	0x10	Tank pressure diagnostics enabled

Data_namn = Software version (e.g. A5COP47T.35G)

Footer information

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

Identifier	**Type**	**Description**
0x01	String	Partnumber
0x02	String	Software ID
0x03	String	Data name
0x04	String	Engine type
0x05	String	Immobilizer code
0x06	---	Unknown
0xFC	ROM end	Last available ROM address
0xFD	ROM offset	Offsett in ROM addresses
0xFE	Code end	Last used ROM address



```
0003ff70h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ; ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0003ff80h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ; ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0003ff90h: FF FF FF FF FF FF 39 31 39 41 36 30 FE 06 30 30 ; ÿÿÿÿÿÿ919A60þ.00
0003ffa0h: 30 30 34 30 FD 06 46 46 46 46 37 30 FC 06 31 30 ; 0040ý.FFFF70ü.10
0003ffb0h: 58 4C 06 04 36 35 32 32 35 31 05 06 20 20 20 20 ; XL..652251..
0003ffc0h: 20 20 20 20 20 20 20 20 20 53 53 56 20 31 43 20 ;          SSV 1C
0003ffd0h: 30 30 30 39 2D 52 34 33 32 42 04 1E 4C 36 33 2E ; 0009-R432B..L63.
0003ffe0h: 4C 4E 31 56 5A 49 35 41 03 0C 37 32 36 30 30 39 ; LN1VZI5A..726009
0003fff0h: 34 02 07 38 39 39 32 30 33 34 01 07 00 A8 E4 96 ; 4..8992034...¨ä–
```

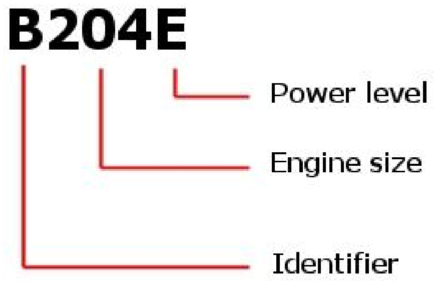
919A60 = Last program address, reversed, actual address is 06A919
000040 = Program size (size of binary), reversed, actual size is 040000
FFFF70 = Flash last address, last available ROM address (07FFFF)
10XL = Unknown value/code (LX10)
652251 = VSS security code, reversed, actual code is 152256
SSV 1C 0009-R432B = Engine type & model, reversed, actual is B234R-9000 1C VSS
L63.LN1VZI5A = Dataname, reversed, actual string is A5IZV1NL.36L
7260094 = Software ID, reversed, actual code is 4900627
8992034 = Partnumber, reversed, actual number is 4302998

Watch out with the dataname, this is also a symbol in the binary, they need not always be the same.

Tuning the T5

Engine types

To be able to tune your engine you first have to know what engine type you have. Saab produced several different engine types with different options.
The engine code will tell you what type it is.



Engine size	Power level	Turbo	Output (m)	Description
204	E/S	Garrett T25	150 bhp [1]	2.0 L Low Pressure Turbo
204	L	Garrett T25	185 bhp	2.0 L High Pressure Turbo
234	E	Garrett T25	170 bhp	2.3 L Low Pressure Turbo
234	L	Garrett T25	200 bhp	2.3 L High Pressure Turbo
234	R	Mitsubishi TD04	225 bhp	2.3 L Aero (TD04 turbo)

[1] 900NG and 9-3 are 154 bhp while 9000 is 150 bhp.

B204E, B204S and B234E

When tuning an LPT engine like the B204E or B234E you will need a BPC valve (Boost Pressure Control) a.k.a. solenoid valve. This valve enables the ECU to control the wastegate on the turbo to control boost levels. The valve comes with three vacuum hoses and should be connected to Wastegate (noted as W on the hoses), Compressor (load side of turbo noted as C on hoses) and Return (resulting unused airflow, noted as R on the hoses). Cars lacking an intercooler should be fitted with one. Optional a boost gauge can be mounted in the car to monitor pressure levels. LPT versions lack a boost gauge (some exceptions are known though). After altering the hardware, replace the LPT software with FPT (B204L or B234L) software and tuning can begin in the same way as described for an FPT engine (B204L, B234L and B234R).

B204L, B234L and B234R

Upgrading cars is often expressed in stages. Stage 1, 2 and 3 are doable without too many costs and hardware modifications. Over stage 3 serious alterations are needed like a larger intercooler, larger turbocharger, larger injectors etc. We will only discuss hardware requirements for stages 1 – 3 here. Stages higher than stage 3 are listed in appendix V.

Stage I

Stage I is achievable with all stock hardware. No modifications needed here. This does not mean that you cannot replace the stock airfilter with an open airfilter, but it is no requirement for stage 1.
Power gain achievable: ~ 20-30 bhp.
Torque gain achievable: ~ 50 Nm.

Stage II

Stage II needs at least a sport exhaust from the catalyst backwards (cat-back). Again an open airfilter is optional but not required.
Power gain achievable: ~ 30-50 bhp.
Torque gain achievable: ~ 80 Nm.

Stage III

Stage III needs at least an open airfilter and a complete sport exhaust (turbo-back). This exhaust must be at least 2.5 inches in diameter. A sport or race catalyst will help spool-up.
Power gain achievable: ~ 60 - 70 bhp.
Torque gain achievable: ~ 100 - 120 Nm.

Tuning in general

To be able to tune the T5 – or any other engine management system for that matter – one has to know about some general issues and terms used in engine management tuning. This chapter will give the reader some understanding of changes taking place inside an engine and how to interpret these while tuning.

General information

To tune an engine means that you optimise your engine for its possibilities. You could be lucky and have the precise right tolerances to make the engine last as long as you want it to. Unfortunately that cannot be achieved during mass production of engines so there will be tolerances that can both be luckily cooperating or opposing each other. These tolerances will decide how much power the engine can produce with safety. Ovalities on the big ends on the connecting rods and the crankshafts bearing surfaces are one possible source for disaster when the engines output rises. Secondly the diameters on the crankshaft's journals are not to be trusted even if the shaft is unused. The bearing shells in the engine block are prone to engine overheating. The shells are deformed and they become both oval and reduced in diameter, bearing failure and crankshaft damage may not be far away! If you are serious about maximising your engine's performance you must make sure that your engine is in good shape. Check that

- The hydraulic valve lifters are quiet a few seconds after engine start.
- The engine does not consume an excessive quantity of oil.
- The engine does not consume any coolant.
- Open the valve cover to get an idea of the engines general condition, check if the oil residue build up is normal compared to the engines mileage and if there are large amounts of it. If this is the case, remove the oil pan and check if the oil strainer is clogged, clean and assemble.
- If the oil strainer is clogged do the procedures described in TSB: 210-2561 utg. 3. (Oil sludge in SAAB 9-5 and 9-3 B205 and B235 engines up to My 2003. Briefly described hereunder:
  - Check that the oil pressure is at least 2,7 bar at 2 000 RPM using 10W/30 oil at 80° C (176° F) engine temperature.
  - Remove the tensioner for the camshaft transmission chain by first removing the small hexagon head with the long spring and small plastic piece at the end. Remove the tensioner and measure that it hasn't protruded more than 12 mm.
  - Measure the bearing play at the crank shaft

Even if your engine is in good shape you take a higher risk of engine failure because of the higher stresses related to the higher power output. The engine manufacturer constructs the engine according to the principle that the average automobile life should be on the far left side of the bell curve of engine failures. If you put more stress to the engine you are moving the bell curve to the left and thus increase the risk of an engine failure.

Consider changing the pistons and the connecting rods if you plan to reach over ~400 Hp.

Hardware requirements

To be able to monitor the engine's condition while tuning and road testing you will need some additional hardware to make sure that you are not actually destroying the engine. This paragraph will list a couple items that you really need to get things right.

- Knock LED. PLEASE install a knock LED so you are alerted when the engine starts to knock! Installation of a knock LED is described in appendix XIII.
- EGT meter. And exhaust gas temperature meter will indicate whether or not the combustions are getting too hot. If this happens, the chance on detonation rises. The maximum allowed temperature for the exhaust gases is about 950 °C. A simple multimeter with a thermocouple can be used as an EGT meter.
- Wideband lambda meter. If you are not getting the AFR information from somewhere else you will need a wideband lambda sonde to monitor the air/fuel ratio.

Knocking

Knocking is a term that means the engine is not igniting in the correct or expected manner. Knocking can be divided in three categories. When Trionic detects knocking and MAP is over 140 kPa it switches to the knocking map (Fuel_knock_mat). When this happens, you actually know that something else in your mapping is not as it should be. Try to prevent the usage of the knocking map as a "knocking handler" but rather see this map as the ultimate fail-safe.

Pinking

Pinking occurs when the ignition is advanced too far for a given environment (engine speed, load etc). The fuel is ignited too early and therefore the pressure of the ignition reaches too high levels before the piston is at TDC. The piston gets a 'blow on the head' and rattles against the bore as it reaches TDC, resulting in a high pitched, mechanical sort of pinging from the engine.
This type of ignition failure is quite common but should be addressed as soon as possible. Engine damage will not occur instantly, but the effect is unwanted from a maintenance point of view as well as from a tuning point of view. Power is of course affected in an undesired way by pinking.

Knocking

Knocking occurs when the 'normally ignited' fuel creates a flame front though the cylinder and the end-gasses (the mixture furthest away from the spark plug) ignite by themselves. This can occur when a fuel with a low octane grade is used.

Detonation / pre-ignition

Detonation or pre-ignition is the worst of the three. This term is used when mixture in the cylinder is ignited by anything else but the spark from the ignition system. This can happen when 'something' (mostly the spark plug itself) gets overheated. The glowing part of the overheated object ignites the fuel at an unwanted moment in time. This will often instantaniously result in severe engine damage.

Volumetric Efficiency (VE)

Normally we would assume that all air taken in by the engine (and thus measured by the ECU) will be burnt in the cylinder. This is very important because this assumption means the amount of fuel to be injected is determined by the amount of air taken in only. In practice this is not the case. Some air will travel though the engine, leaving the engine though the exhaust. Some air will even travel back into the air intake. This effect is known as the volumetric efficiency of the engine. To compensate for the air measured but not burnt in the engine, correction maps have been introduced. These are known as VE maps.

PID Control

http://en.wikipedia.org/wiki/PID_controller

A PID controller is a feedback control system that tries to get an output value for the system equal to a reference value. It uses a very complex mathematical algorithm to do so which will be described in this chapter. At any time I will try to give practical information to make the chapter more understandable. The major part that we are concerned about when tuning a car – as far as PID controllers are involved – is the boost regulation (charging pressure). When boost is requested – by opening the throttle – the system detects that there is a difference between the requested boost and the actual boost in the inlet manifold. The example given throughout this chapter will be idling and then requesting maximum boost pressure (at wide open throttle). The error that the PID controller has to correct is maximal at this point and also includes the problem of overshooting.
There are 6 major values to consider when looking at the PID controller.

1. The actual boost pressure: the boost pressure as measured through the MAP sensor.
2. The requested boost pressure: the value that has been fetched from the boost request table (Tryck_mat or Tryck_mat_a).
3. The controller's calculated output
4. The controller's P value (proportional gain)
5. The controller's I value (integral gain)
6. The controller's D value (derivative gain)

Since 1 and 2 are quite obvious we will not discuss them any further here.

Controller's calculated output

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). So the amount of correction taking place on the boost pressure depends on three factors: Pout, Iout and Dout.

$$\mathrm{MV}(\mathrm{t}) = P_{\mathrm{out}} + I_{\mathrm{out}} + D_{\mathrm{out}}$$

where Pout, Iout, and Dout are the contributions to the output from the PID controller from each of the three terms, as defined below. The PID's output value is simply a number which indicates what PWM signal to apply to the solenoid valve to correct the difference between requested boost and measured boost.

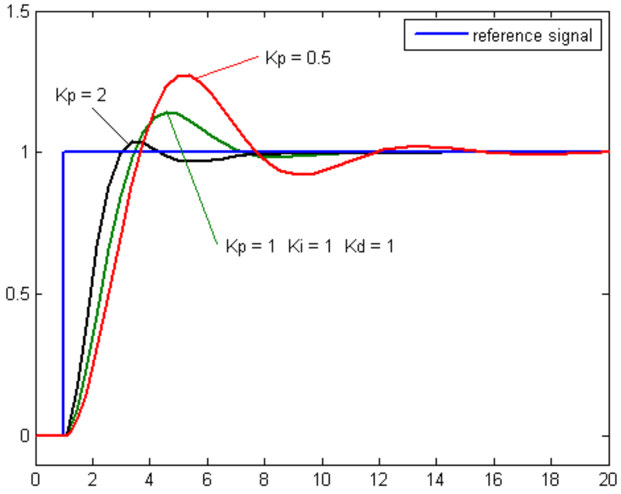
Proportional output (Pout)

The proportional gain makes a change to the controller's output that is proportional to the current difference between measured boost and boost request. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain.
The proportional term is given by:

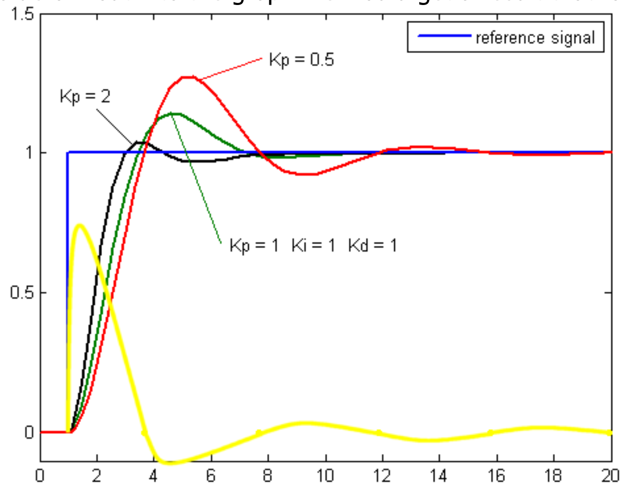
$$P_{\text{out}} = K_p\, e(t)$$

Pout: proportional output
Kp: proportional gain fetched from P_fors table which depends on RPM and boost error.
e: error = boost requested minus boost measured
t: Time or instantaneous time (the present)

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive (or sensitive) controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances. In the absence of disturbances, pure proportional control will not settle at its target value, but will retain a steady state error that is a function of the proportional gain and the process gain. Despite the steady-state offset, both tuning theory and industrial practice indicate that it is the proportional term that should contribute the bulk of the output change.
In more detail the P factor has a result as shown in the image below.



As we can see, the P factor determines the steepness of the inclination of the curve. In terms of throttle response we would like to have P as high as possible without the system overshooting too much or becoming unstable (hunting for requested boost pressure).
If we try to draw the value of Pout into the graph we would get a result that looks like the yellow line.



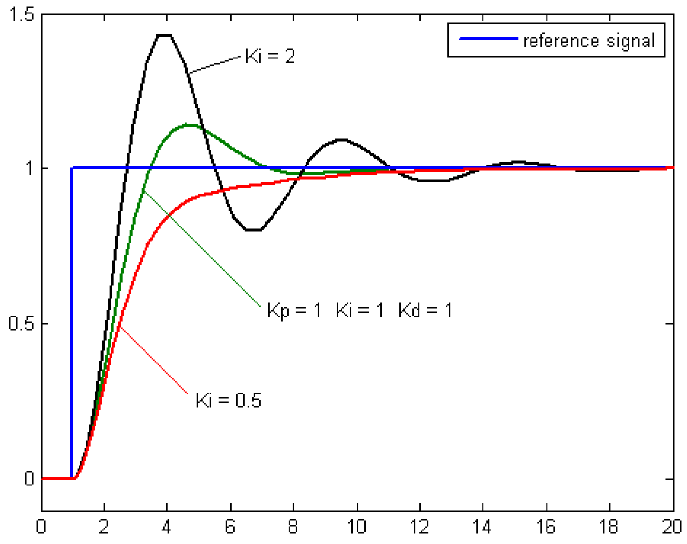
Integral output (Iout)

The contribution from the integral term is proportional to both the magnitude of the current error (requested boost minus measured boost) and the duration of the error. Summing the instantaneous error over time (integrating the error) gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain which is fetched from the I_fors table) and added to the controller's output. So, the influence of the integral factor depends on the integral gain from I_fors. The fetched value is called Ki.

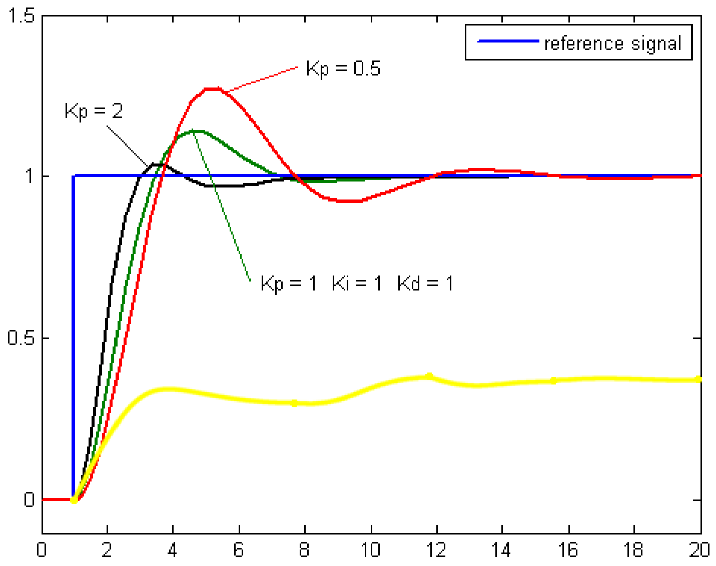
$$I_{\mathrm{out}} = K_i \int_0^t e(\tau)\, d\tau$$

Iout: Integral output
Ki: integral gain fetched from I_fors which depends on RPM and boost error
e: error = boost requested minus boost measured
τ: time in the past contributing to the integral response

The integral term (when added to the proportional term) accelerates the movement of the process towards the requested boost and eliminates the residual steady-state error that occurs with a proportional only controller. However, since the integral term is responding to accumulated errors from the past, it can cause the present value to overshoot the requested boost value (create an error in the other direction so that measured boost will be higher than the requested boost).



If we draw the Iout value (indicative again) in the image we would get the yellow line in this image.



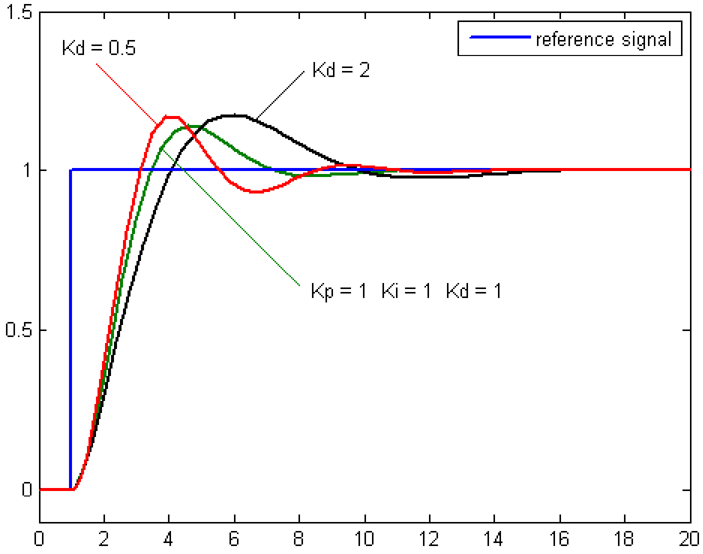
Derivative output (Dout)

The rate of change of the boost pressure error is calculated by determining the slope of the error over time (i.e. its first derivative with respect to time) and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, Kd.

$$D_{\text{out}} = K_d \frac{de}{dt}$$

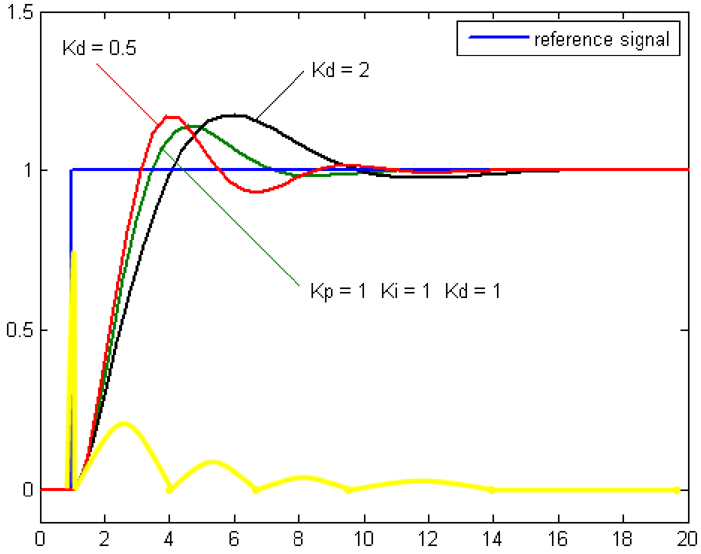
Dout: Derivative output
Kd: Derivative Gain, a tuning parameter
e: Error = SP − PV
t: Time or instantaneous time (the present)

The derivative term slows the rate of change of the controller output and this effect is most noticeable close to the controller set point. Hence, derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability. However, differentiation of a signal amplifies noise and thus this term in the controller is highly sensitive to noise in the error term, and can cause a process to become unstable if the noise and the derivative gain are sufficiently large.



If we look at the graph above we can see that the error starts off as zero (engine = idling). The moment the boost request changes (throttle opened) The error changes very fast to 1 (blue line). The change rate is highest at this point. Where the red line is flat, there is no change in error so Dout will be zero there. If we look at the next image we can see the rate of change in the yellow line.

We now see that *Dout* is only a factor when boost inclines or declines very fast. This is shown in the yellow line (indicative) in the image below.



PID controller output

The output from the three terms, the proportional, the integral and the derivative terms are summed to calculate the output of the PID controller. Defining u(t) as the controller output, the final form of the PID algorithm is:

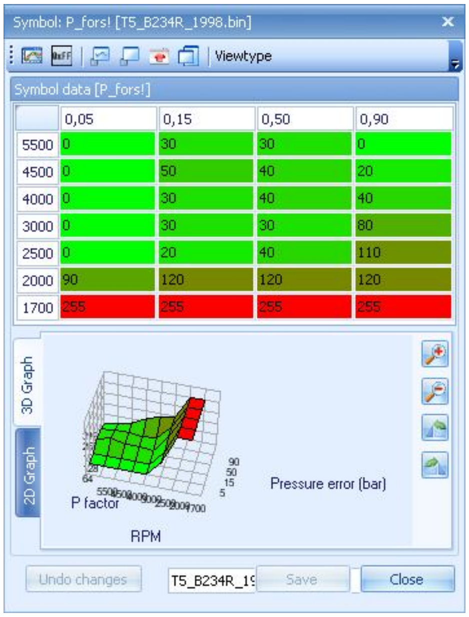
$$\mathrm{u(t)} = \mathrm{MV(t)} = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{de}{dt}$$

- Kp: Proportional Gain - Larger Kp typically means faster response since the larger the error, the larger the Proportional term compensation. An excessively large proportional gain will lead to process instability and oscillation.
- Ki: Integral Gain - Larger Ki implies steady state errors are eliminated quicker. The trade-off is larger overshoot: any negative error integrated during transient response must be integrated away by positive error before we reach steady state.
- Kd: Derivative Gain - Larger Kd decreases overshoot, but slows down transient response and may lead to instability due to signal noise amplification in the differentiation of the error.

Effects of *increasing* parameters				
Parameter	Rise Time	Overshoot	Settling Time	Offset Error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Decrease	Decrease	Decrease	None

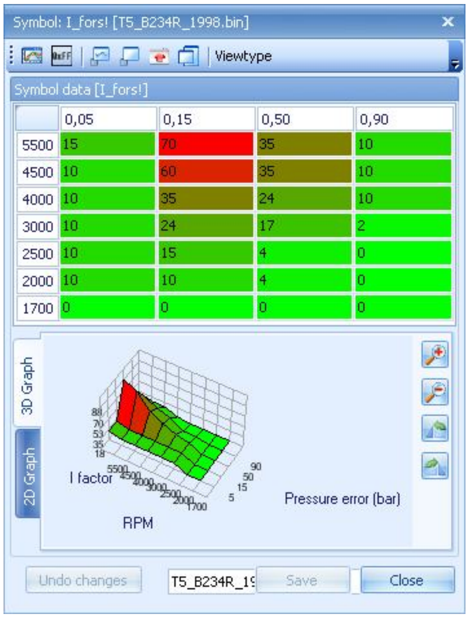
Implementation in Trionic 5

If we take a closer look at the parameters in the Trionic software we can see that the factors Kp, Ki and Kd depend on RPM and pressure error. This makes the already complicated system even more complex because not only is the boost request and measured boost a variable but also the factors in which P, I and D are applied. This makes tuning these parameter tables very difficult.



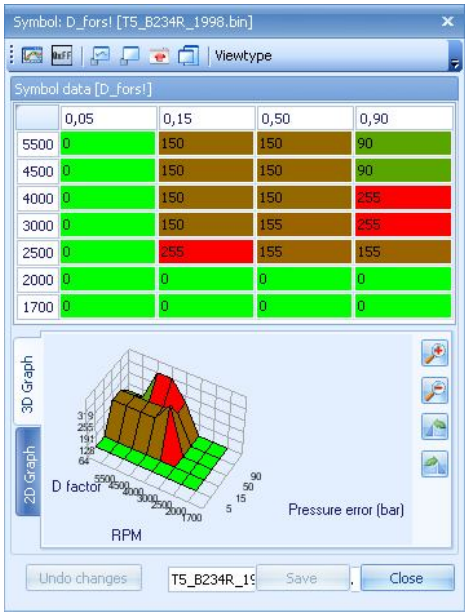
	0,05	0,15	0,50	0,90
5500	0	30	30	0
4500	0	50	40	20
4000	0	30	40	40
3000	0	30	30	80
2500	0	20	40	110
2000	90	120	120	120
1700	255	255	255	255

The image above shows a stock P_fors (Kp in out previous paragraphs) table for manual transmission on a B234R engine. We can clearly see that values differ a lot with RPM and pressure error. Low pressure errors (0-0.10 bar) are not adjusted over 2500 RPM. The highest Kp is found in the lowest RPM range. This is probably because the turbo still needs to spool up on this region so regulation is strong here.



	0,05	0,15	0,50	0,90
5500	15	70	35	10
4500	10	60	35	10
4000	10	35	24	10
3000	10	24	17	2
2500	10	15	4	0
2000	10	10	4	0
1700	0	0	0	0

Here we see the Ki factors used in the stock firmware. What seems kind of strange is that I peaks at a low boost error and high range RPM.



	0,05	0,15	0,50	0,90
5500	0	150	150	90
4500	0	150	150	90
4000	0	150	150	255
3000	0	150	155	255
2500	0	255	155	155
2000	0	0	0	0
1700	0	0	0	0

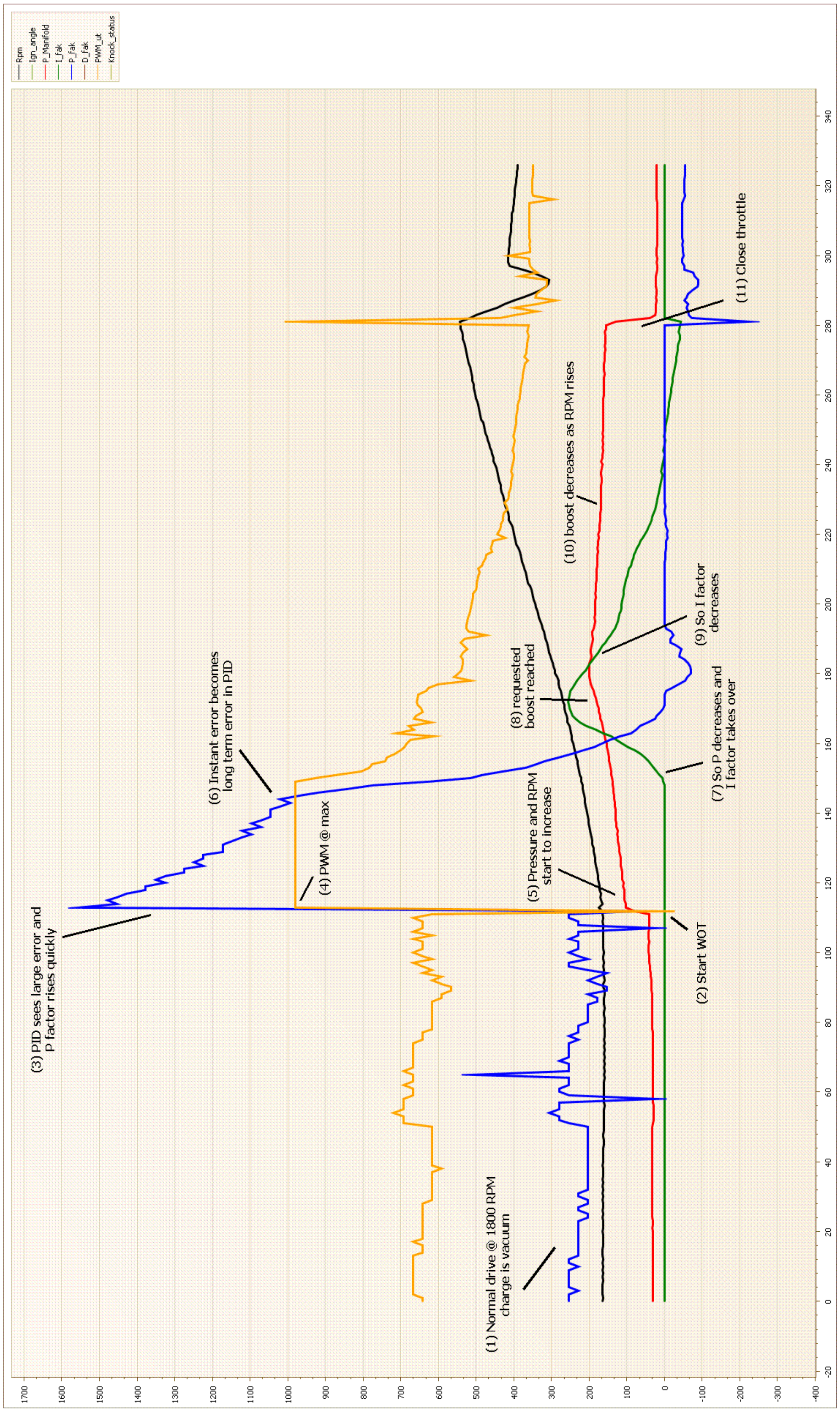
And finally we take a look at the Kd factor in the stock firmware. It seems kind of dumb to have these strange high values in the table at seemingly random places. The fact that Kd is low on low error values is quite understandable because we saw earlier that Kd is only an active component when boost pressure changes quickly.

If we step back to our example of idling and then opening the throttle 100%, we can see that the boost error will be large at low RPM. Looking at the tables we find that Kp starts of high, Ki is zero and Kd is zero. The system will start to work its way to the requested boost. While it is doing so, RPM rises and the boost error decreases. Suppose we pass 2500 RPM and the boost error has decreased to 0.5 bar. We now have a Kp of 40, Ki of 4 and a Kd of 155. So, the system is suddenly completely different and will not increase boost so fast anymore because Kp is much lower. Kd is coming into play to prevent overshoot and Ki is rising to settle to the right requested boost. We just keep going on WOT and our RPM has climbed up to 4000 RPM and the pressure error is virtually non existing. 0.05 bar. We now have a Kp of 0, an Ki of 10 and a Kd of 0. Again the entire system has changed and we now have a situation where boost inclination is very low and we are just keeping at this pressure level.

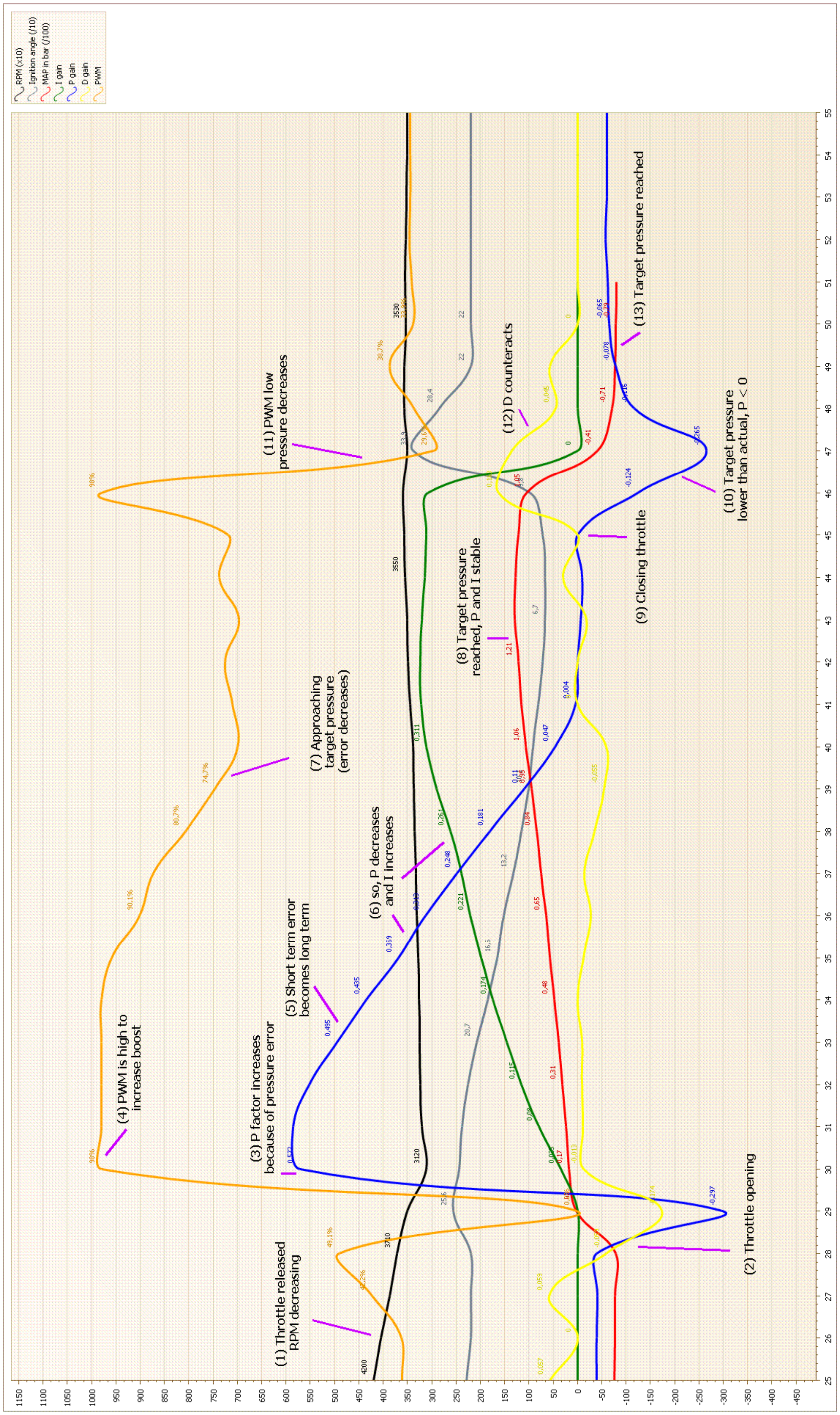
As you can see, getting familiar with the PID control algorithm is a very time consuming factor. If you don't have to alter these settings, please don't. **You must know exactly what you are doing and exactly what your goals** are before you change the PID control parameters. More boost doesn't require the PID control parameters to be changed, just change the boost request maps and the boost regulation control tables (reg_kon_mat tables) for that.

Having a drive

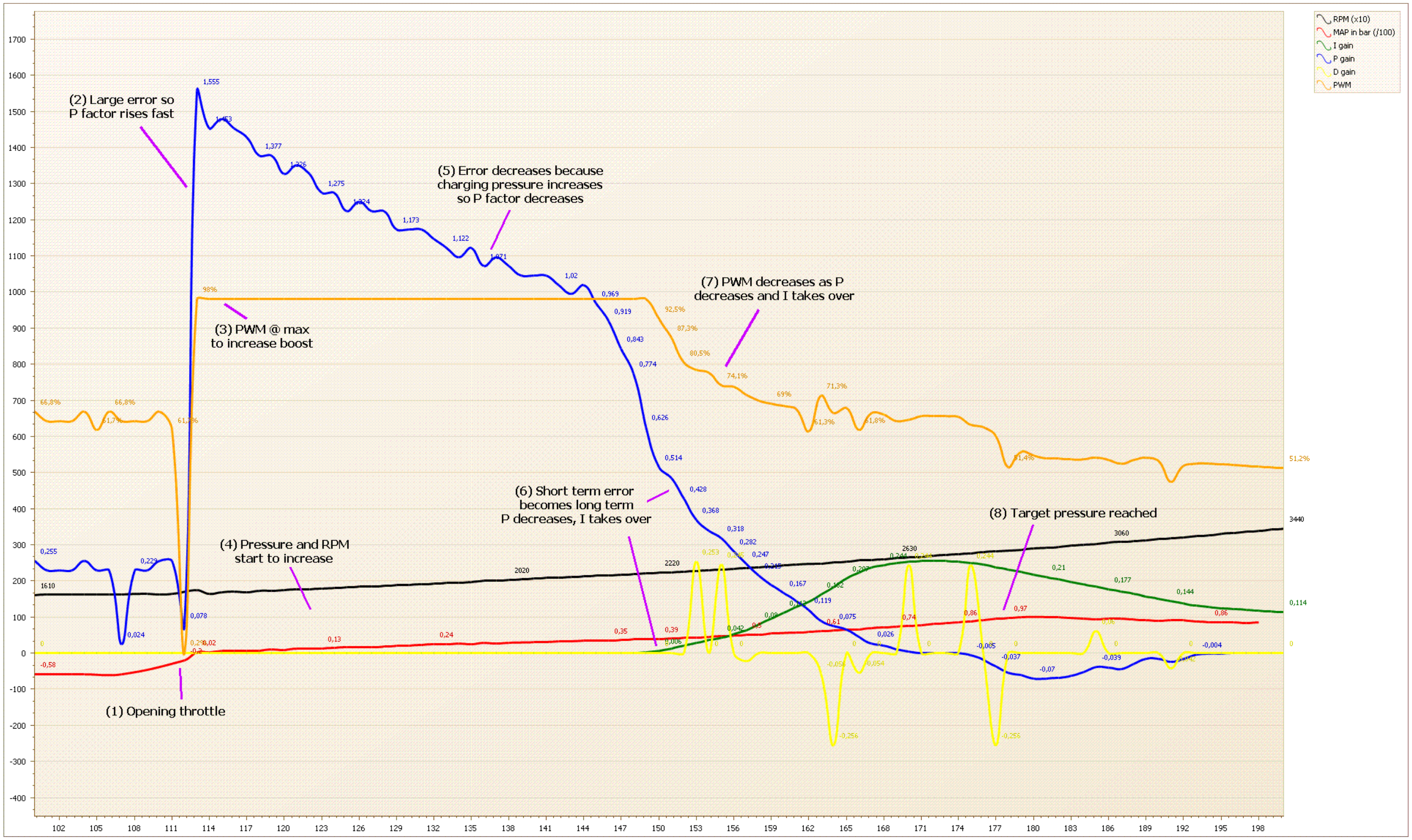
If we drive the car and log RPM, Manifold pressure, P_factor, I_factor and PWM signal we end up with a graph like shown below.



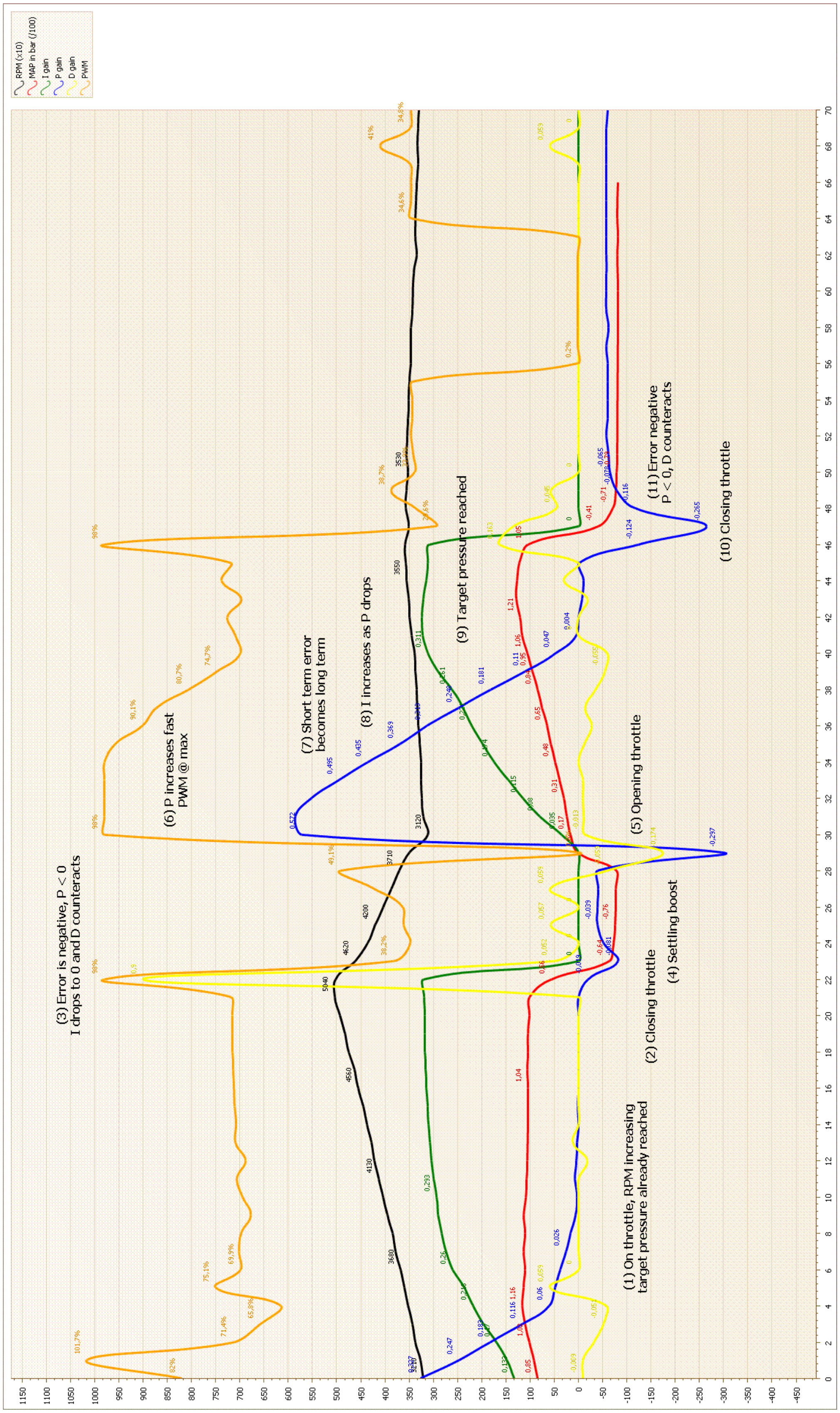
In more detail, throttle response in relation to PID parameters and PWM output can be read from this image.



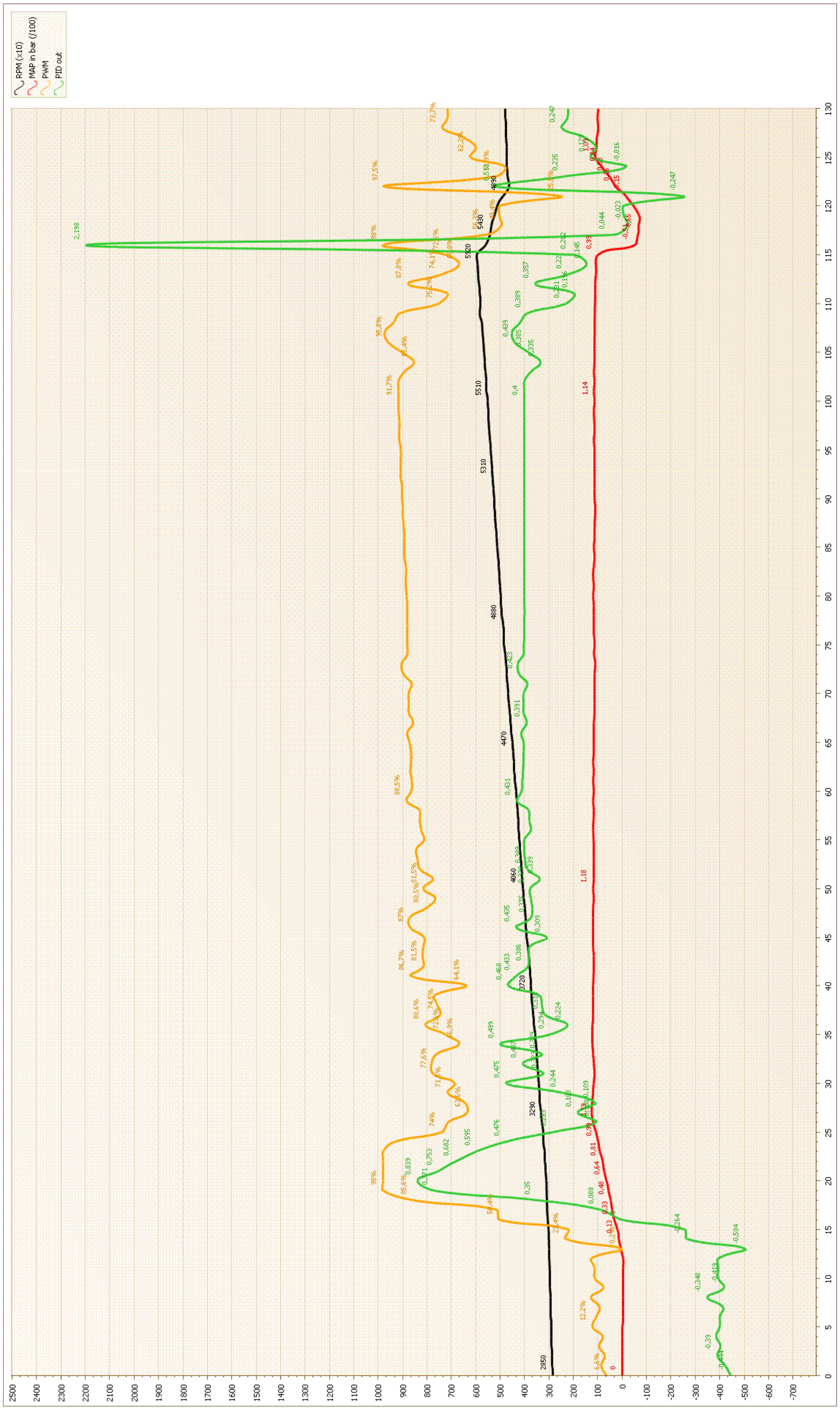
If we WOT the car under load the response can be as shown in this image.



Closing and opening the throttle in respect to the PID/PWM parameters are shown in this last image.



If we add Pout, Iout and Dout and plot it against the PWM output we can see that there is a very strong resemblance.



Reading the code

Reading the assembler code from the Trionic gives us more insight about the PID behaviour and lets us verify the things we see in practice. The major formula for the calculation of the PWM signal on the boost control valve is:

$$PWM\_ut = APC\_Knock * (reg\_kon\_mat + temperature\_correction + P\_gain + I\_Gain + D\_Gain + Apc\_Adapt)$$

In which

APC knock

From this we see that knocking can decrease charging pressure (we already knew this, but the formula confirms it) with a certain factor.
APC_Knock = (100 - APC_decrese) / 100
Where APC_decrese = factor that determines how much boost pressure should be lowered in case of knocking

Reg_kon_mat

Reg_kon_mat is an offset table that determines what offset to add to the PID controllers output to get to the PWM output signal. We can see this in the previous image… the difference in offset between the yellow and the green line is the current reg_kon_mat value.
The percentage found in reg_kon_mat is actually the bias (preset) of PWM signal in a given RPM site.

Temperature_correction

Temperature	Addition to PWM_ut
Below 0 ˚C	0
Between 0 ˚C and 40 ˚C	Temperature * 2
Over 40 ˚C	80

P_gain

$$P\_gain = P\_factor * (pressure\_error / 10)$$

Where pressure error = 50 when 0.5 bar

I_gain

$$I\_gain = I\_factor * (cumulative\_pressure\_error / 10)$$

Where cumulative pressure error = I_factor * pressure_error + previous cumulative error
The cumulative pressure error gets cleared (set to zero) whenever:
The cumulative pressure error > 100 and I_factor > I_fak_max
The cumulative pressure error < -100 and I_factor < -I_fak_max
on NOT wide open throttle I_gain = 0
on break I_gain = 0
on cruise control I_gain = 0

D_gain

$$D\_gain = (D\_factor * pressure\_error) / 2$$

Where pressure error = 50 when 0.5 bar

Notes

Under 1000 rpm PWM_ut = 0.2 %
PWM_ut maximum = 98%
PWM_ut minimum = 0.2%

Tuning with T5Suite

T5Suite incorporates several ways to tune your software to another stage.

Manual tuning

You can alter maps and parameters within the binary file using T5Suite. This is done by selecting the desired symbol from the symbollist and double clicking it. A mapviewer will appear and you can manually alter the values in the map.

Map presentation

You can view the data in a map in several ways: hexadecimal, decimal and easy.
In hexadecimal mode the numbers in the map are displayed as hexadecimal numbers (0-F). If you don't know how to interpret hexadecimal numbers you can switch the view to decimal numbers (0-9) or even easy view. Easy view lets you view the map in understandable numbers for as much as the maps are known. Boost pressure maps for example are converted from numbers ranging from 0 to 255 to actual boost pressure values (-1 bar to 1.55 bar).

Color indicators

The map can be displayed in table form as well as in graphical form. The values can (depending on settings made by the user) be displayed in colors ranging either from transparent to red or from green to red. The coloring can be disabled altogether for performance reasons.

Altering values in a map

To edit a value in a map just point and click on the specific cell. To edit several cells at once you can hold the Ctrl-button on your keyboard then point and click on each individual cell you want to edit the same way. Finally you can click, hold and drag to select several cells in the same box. To avoid that you have to adjust all values of a "large" map manually some features have been added to the map editor:

- Plus key: adds 1 to all selected cells
- Minus key: subtracts 1 from all selected cells
- PageUp key: adds 10 to all selected cells
- PageDown key: subtracts 10 from all selected cells
- Home key: sets all selected cells to the maximal value
- End key: sets all selected cells to the minimal values

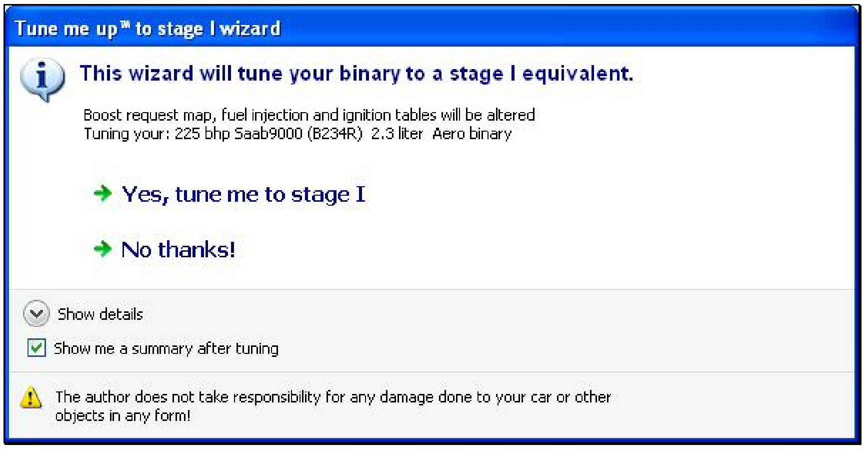
And a mathematics tab where you can select:

- A value to fill the selected cells with.
- A value to add to the selected cells.
- A value to multiply the selected cells by.
- A value to divide the selected cells by.

To execute what you have selected in this tab, push the lightning button.
You can also select one or multiple cells, right click and chose "copy selected cells" or "Paste selected cells" where you could choose: "At original position" or "At currently selected position".

Tuning wizard

T5Suite can tune most binaries automatically. Stages I, II, III are standard. After selecting a stage a wizard will appear. Please check appendix V for hardware requirements.



Tune me up™ to stage I wizard

This wizard will tune your binary to a stage I equivalent.

Boost request map, fuel injection and ignition tables will be altered
Tuning your: 225 bhp Saab9000 (B234R) 2.3 liter Aero binary

➜ Yes, tune me to stage I

➜ No thanks!

Show details

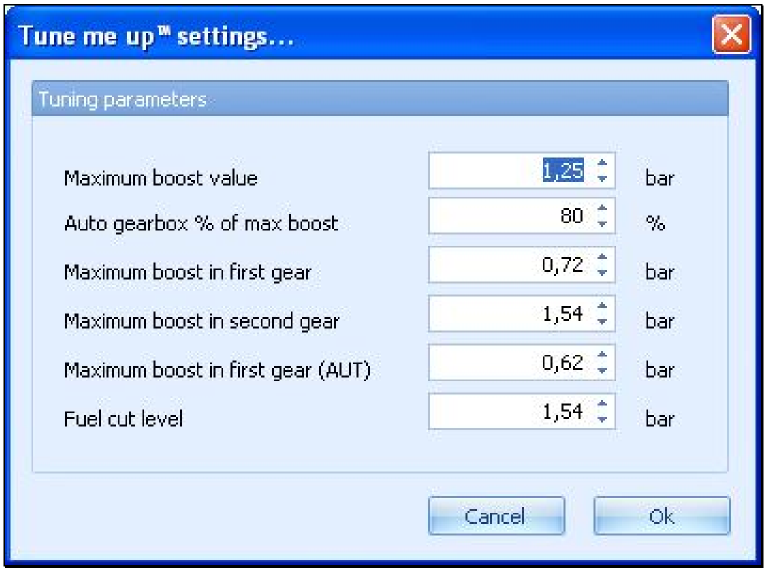
☑ Show me a summary after tuning

The author does not take responsibility for any damage done to your car or other objects in any form!

When the user selects to continue with the wizard, adaptions will be made to boost request tables, injection table, knocking map tables etc. After making the changes a report will be generated indicating exactly which changes have been made to your software. Of course a backup of your binary is created before making the changes.

Stage	Boost B234E/L	Boost B234R, B204E/L/R	Boost B234E/L (AUT)	Boost B234R, B204E/L/R (AUT)
I	1.05	1.15	0.95	1.04
II	1.15	1.25	1.04	1.13
III	1.25	1.35	1.13	1.22

Tuning to stage X enables you to set the parameters for the wizard manually.



Tune me up™ settings...

Tuning parameters

Maximum boost value	1,25	bar
Auto gearbox % of max boost	80	%
Maximum boost in first gear	0,72	bar
Maximum boost in second gear	1,54	bar
Maximum boost in first gear (AUT)	0,62	bar
Fuel cut level	1,54	bar

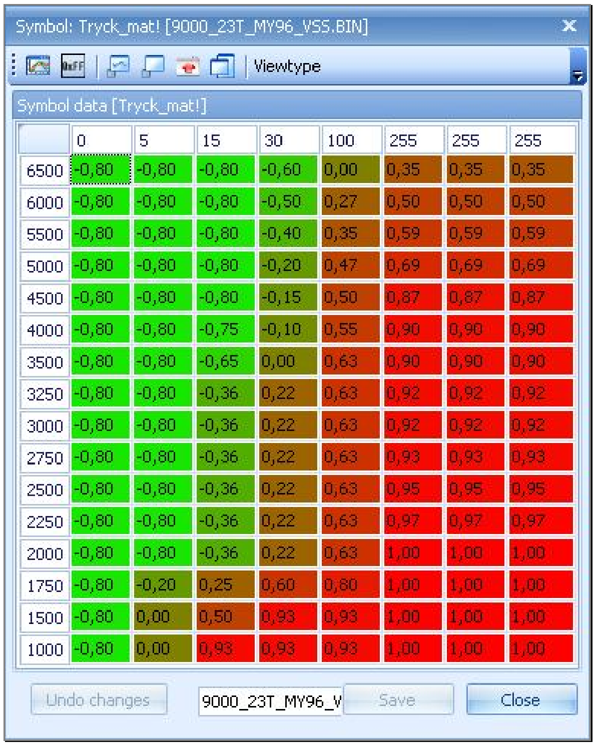
Converting to 3 bar mapsensor

Introduction

This chapter will explain the issues that one will encounter when the need for a 3 bar mapsensor rises. This need surfaces when a tuned car must reach over ~1.4 bar boost pressure. The standard mapsensor is a 2.5 bar variant and hence will measure correctly to approx. 2.5 bar minus environment pressure (which is ~1 bar) minus saturation level of the sensor ~3% so about 1.4 bar.
Above this boost pressure level the standard mapsensor will not suffice anymore and an upgraded one should be mounted for the Trionic to be able to measure boost pressure correctly.

Issues

The real issues we're faced with when doing this conversion are not of the hardware type.
Boost pressure calculation is one of the smaller issues that we encounter. Suppose we are working on a binary that has the following boost request map (Tryck_mat).



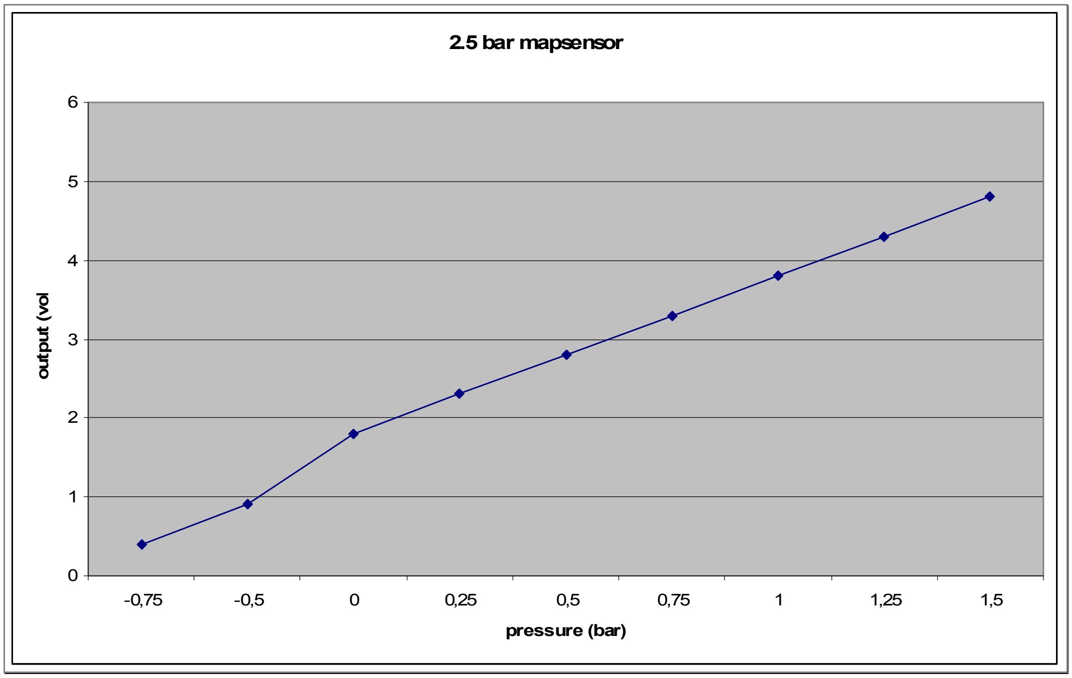
	0	5	15	30	100	255	255	255
6500	-0,80	-0,80	-0,80	-0,60	0,00	0,35	0,35	0,35
6000	-0,80	-0,80	-0,80	-0,50	0,27	0,50	0,50	0,50
5500	-0,80	-0,80	-0,80	-0,40	0,35	0,59	0,59	0,59
5000	-0,80	-0,80	-0,80	-0,20	0,47	0,69	0,69	0,69
4500	-0,80	-0,80	-0,80	-0,15	0,50	0,87	0,87	0,87
4000	-0,80	-0,80	-0,75	-0,10	0,55	0,90	0,90	0,90
3500	-0,80	-0,80	-0,65	0,00	0,63	0,90	0,90	0,90
3250	-0,80	-0,80	-0,36	0,22	0,63	0,92	0,92	0,92
3000	-0,80	-0,80	-0,36	0,22	0,63	0,92	0,92	0,92
2750	-0,80	-0,80	-0,36	0,22	0,63	0,93	0,93	0,93
2500	-0,80	-0,80	-0,36	0,22	0,63	0,95	0,95	0,95
2250	-0,80	-0,80	-0,36	0,22	0,63	0,97	0,97	0,97
2000	-0,80	-0,80	-0,36	0,22	0,63	1,00	1,00	1,00
1750	-0,80	-0,20	0,25	0,60	0,80	1,00	1,00	1,00
1500	-0,80	0,00	0,50	0,93	0,93	1,00	1,00	1,00
1000	-0,80	0,00	0,93	0,93	0,93	1,00	1,00	1,00

Image 11: Standard FPT boost request map

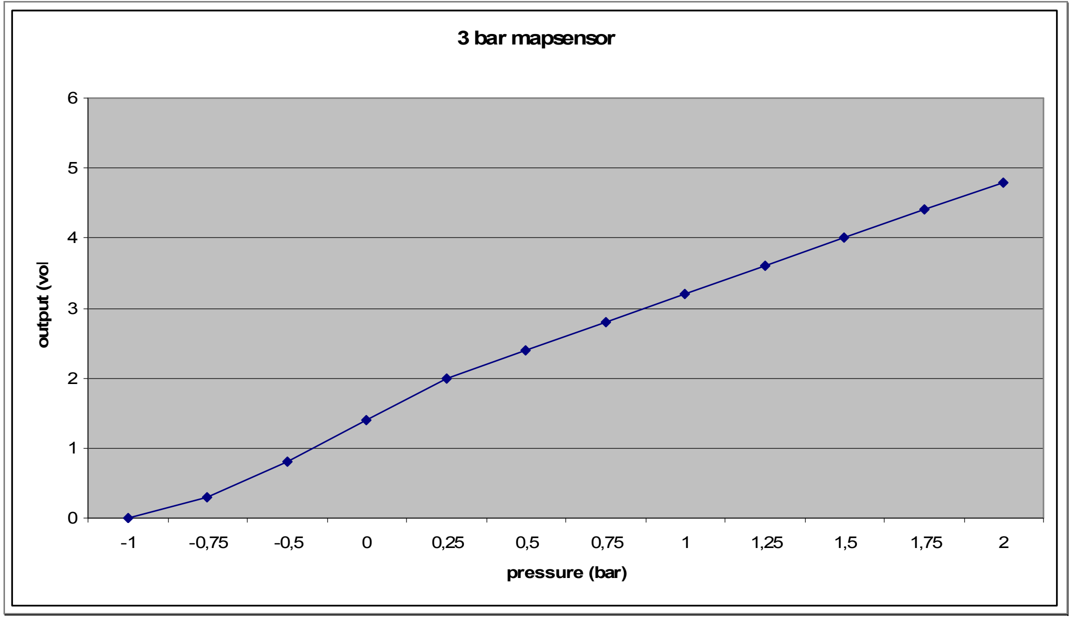
The values displayed are the requested boost pressure for a given throttle position and rpm. The boost requested at 2000 rpm at wide open throttle (WOT) is 1.00 bar. Trionic will try to reach this requested boost level by controlling the boost control valve (solenoid) using a complex control mechanism.

Suppose we hook up this car with a 3 bar mapsensor and change nothing else.
The Trionic would then measure lower voltages from the sensor for a given boost level.

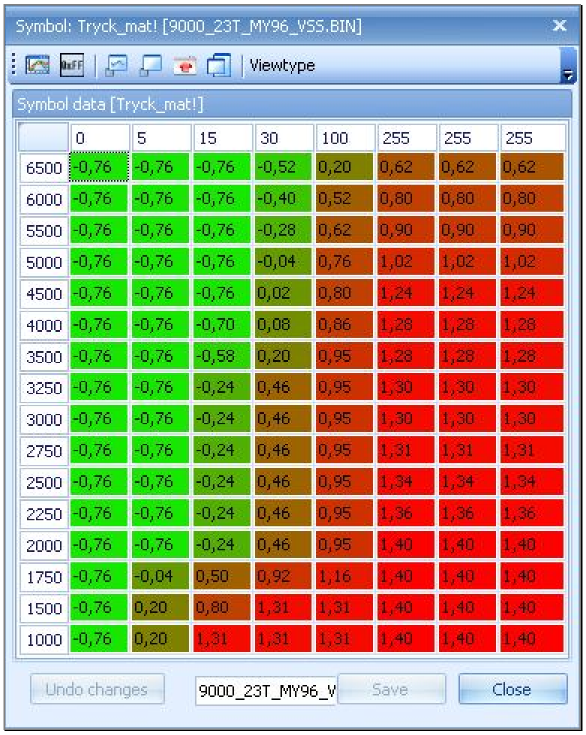
The output voltage values for the stock 2.5 bar mapsensor are:



For the 3 bar sensor these are the approximate values:



Hence, we could change the map to the boost levels the Trionic would be trying to reach.



Symbol: Tryck_mat! [9000_23T_MY96_VSS.BIN]

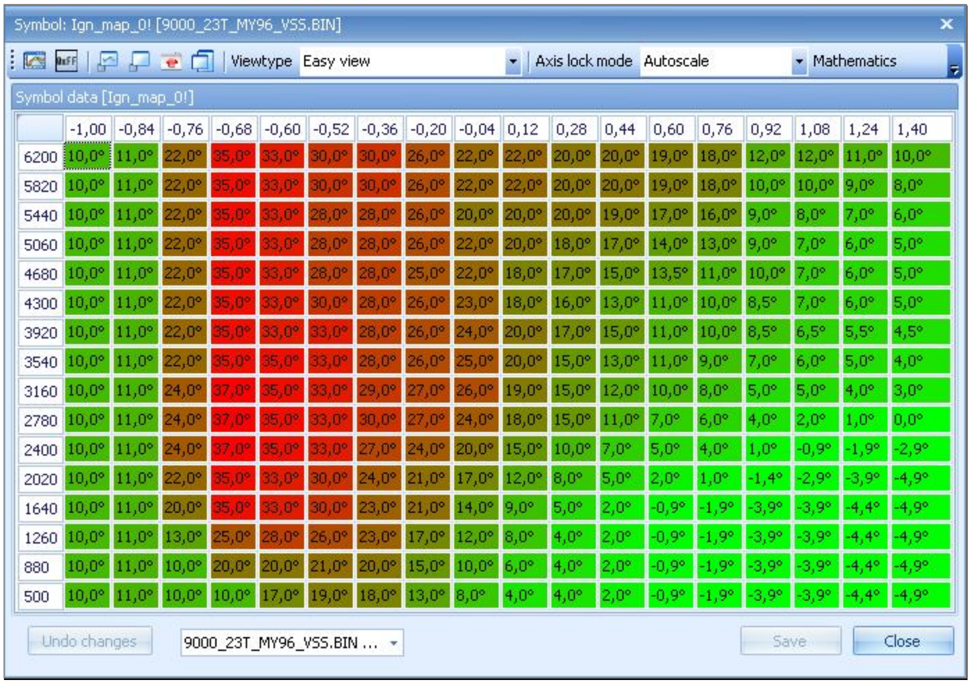
Symbol data [Tryck_mat!]

	0	5	15	30	100	255	255	255
6500	-0,76	-0,76	-0,76	-0,52	0,20	0,62	0,62	0,62
6000	-0,76	-0,76	-0,76	-0,40	0,52	0,80	0,80	0,80
5500	-0,76	-0,76	-0,76	-0,28	0,62	0,90	0,90	0,90
5000	-0,76	-0,76	-0,76	-0,04	0,76	1,02	1,02	1,02
4500	-0,76	-0,76	-0,76	0,02	0,80	1,24	1,24	1,24
4000	-0,76	-0,76	-0,70	0,08	0,86	1,28	1,28	1,28
3500	-0,76	-0,76	-0,58	0,20	0,95	1,28	1,28	1,28
3250	-0,76	-0,76	-0,24	0,46	0,95	1,30	1,30	1,30
3000	-0,76	-0,76	-0,24	0,46	0,95	1,30	1,30	1,30
2750	-0,76	-0,76	-0,24	0,46	0,95	1,31	1,31	1,31
2500	-0,76	-0,76	-0,24	0,46	0,95	1,34	1,34	1,34
2250	-0,76	-0,76	-0,24	0,46	0,95	1,36	1,36	1,36
2000	-0,76	-0,76	-0,24	0,46	0,95	1,40	1,40	1,40
1750	-0,76	-0,04	0,50	0,92	1,16	1,40	1,40	1,40
1500	-0,76	0,20	0,80	1,31	1,31	1,40	1,40	1,40
1000	-0,76	0,20	1,31	1,31	1,31	1,40	1,40	1,40

Image 12: Standard boost request map in 3 bar view

Now we can clearly see that the boost levels the Trionic would be trying to reach are way higher. But… now we come to the real issues…

Imagine the ignition map having boost pressure as x axis. Normally the most ignition advance will be around a certain boost pressure level.



Symbol: Ign_map_0! [9000_23T_MY96_VSS.BIN]

Viewtype Easy view

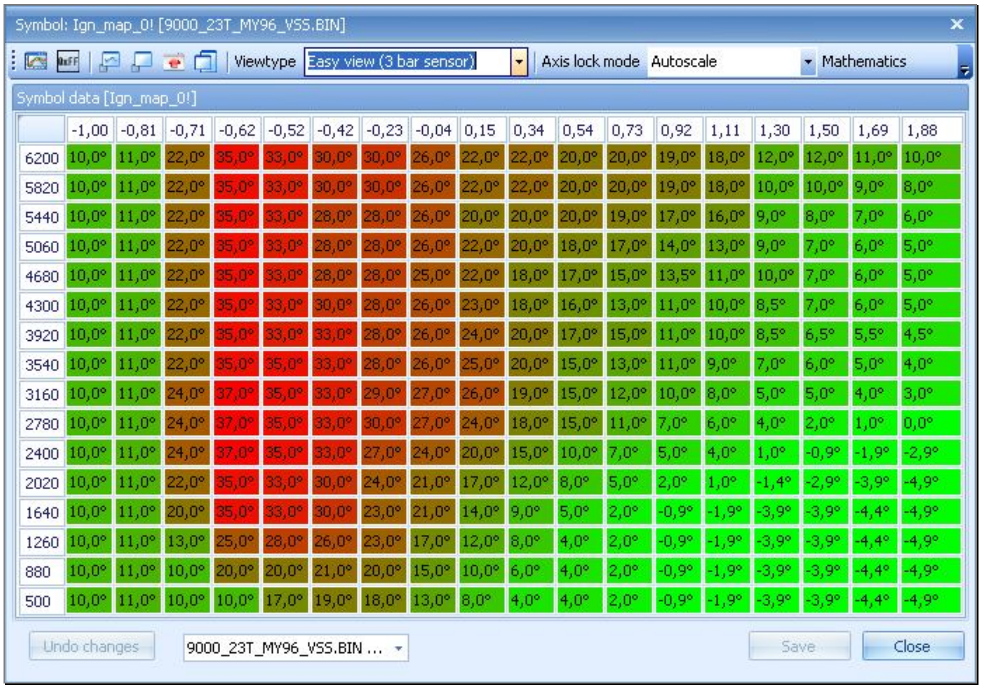
Axis lock mode Autoscale

Symbol data [Ign_map_0!]

	-1,00	-0,84	-0,76	-0,68	-0,60	-0,52	-0,36	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24	1,40
6200	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	30,0°	26,0°	22,0°	22,0°	20,0°	20,0°	19,0°	18,0°	12,0°	12,0°	11,0°	10,0°
5820	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	30,0°	26,0°	22,0°	22,0°	20,0°	20,0°	19,0°	18,0°	10,0°	10,0°	9,0°	8,0°
5440	10,0°	11,0°	22,0°	35,0°	33,0°	28,0°	28,0°	26,0°	20,0°	20,0°	20,0°	19,0°	17,0°	16,0°	9,0°	8,0°	7,0°	6,0°
5060	10,0°	11,0°	22,0°	35,0°	33,0°	28,0°	28,0°	26,0°	22,0°	20,0°	18,0°	17,0°	14,0°	13,0°	9,0°	7,0°	6,0°	5,0°
4680	10,0°	11,0°	22,0°	35,0°	33,0°	28,0°	28,0°	25,0°	22,0°	18,0°	17,0°	15,0°	13,5°	11,0°	10,0°	7,0°	6,0°	5,0°
4300	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	28,0°	26,0°	23,0°	18,0°	16,0°	13,0°	11,0°	10,0°	8,5°	7,0°	6,0°	5,0°
3920	10,0°	11,0°	22,0°	35,0°	33,0°	33,0°	28,0°	26,0°	24,0°	20,0°	17,0°	15,0°	11,0°	10,0°	8,5°	6,5°	5,5°	4,5°
3540	10,0°	11,0°	22,0°	35,0°	35,0°	33,0°	28,0°	26,0°	25,0°	20,0°	15,0°	13,0°	11,0°	9,0°	7,0°	6,0°	5,0°	4,0°
3160	10,0°	11,0°	24,0°	37,0°	35,0°	33,0°	29,0°	27,0°	26,0°	19,0°	15,0°	12,0°	10,0°	8,0°	5,0°	5,0°	4,0°	3,0°
2780	10,0°	11,0°	24,0°	37,0°	35,0°	33,0°	30,0°	27,0°	24,0°	18,0°	15,0°	11,0°	7,0°	6,0°	4,0°	2,0°	1,0°	0,0°
2400	10,0°	11,0°	24,0°	37,0°	35,0°	33,0°	27,0°	24,0°	20,0°	15,0°	10,0°	7,0°	5,0°	4,0°	1,0°	-0,9°	-1,9°	-2,9°
2020	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	24,0°	21,0°	17,0°	12,0°	8,0°	5,0°	2,0°	1,0°	-1,4°	-2,9°	-3,9°	-4,9°
1640	10,0°	11,0°	20,0°	35,0°	33,0°	30,0°	23,0°	21,0°	14,0°	9,0°	5,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°
1260	10,0°	11,0°	13,0°	25,0°	28,0°	26,0°	23,0°	17,0°	12,0°	8,0°	4,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°
880	10,0°	11,0°	10,0°	20,0°	20,0°	21,0°	20,0°	15,0°	10,0°	6,0°	4,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°
500	10,0°	11,0°	10,0°	10,0°	17,0°	19,0°	18,0°	13,0°	8,0°	4,0°	4,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°

Image 13: Standard ignition map

Now let's change the boost pressure axis to the values the Trionic would be measuring when a 3 bar mapsensor would be installed.



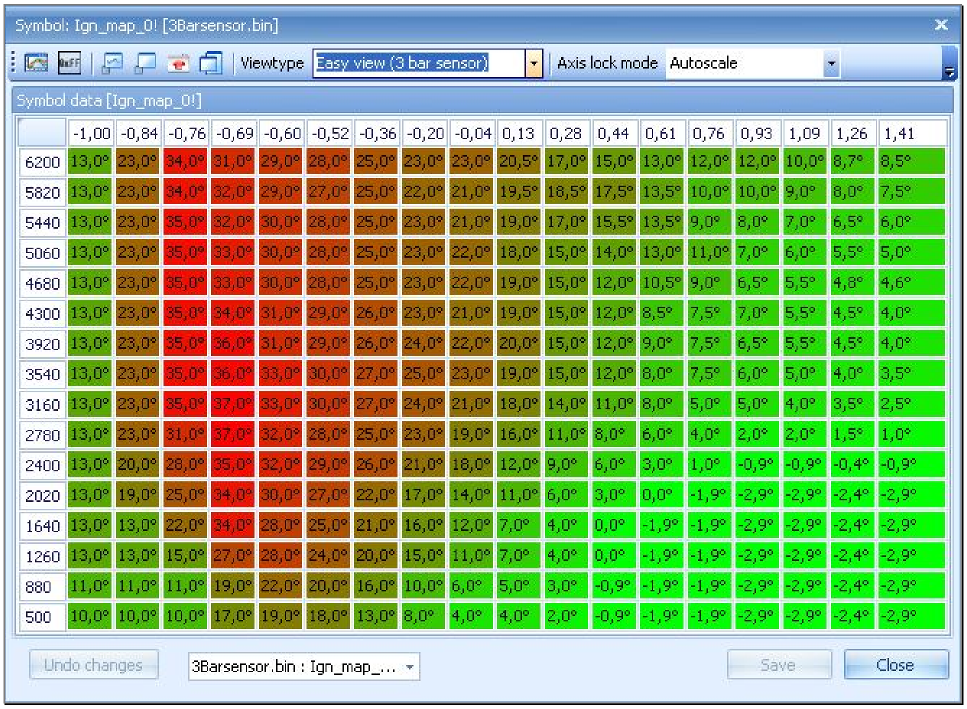
	-1,00	-0,81	-0,71	-0,62	-0,52	-0,42	-0,23	-0,04	0,15	0,34	0,54	0,73	0,92	1,11	1,30	1,50	1,69	1,88
6200	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	30,0°	26,0°	22,0°	22,0°	20,0°	20,0°	19,0°	18,0°	12,0°	12,0°	11,0°	10,0°
5820	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	30,0°	26,0°	22,0°	22,0°	20,0°	20,0°	19,0°	18,0°	10,0°	10,0°	9,0°	8,0°
5440	10,0°	11,0°	22,0°	35,0°	33,0°	28,0°	28,0°	26,0°	20,0°	20,0°	20,0°	19,0°	17,0°	16,0°	9,0°	8,0°	7,0°	6,0°
5060	10,0°	11,0°	22,0°	35,0°	33,0°	28,0°	28,0°	26,0°	22,0°	20,0°	18,0°	17,0°	14,0°	13,0°	9,0°	7,0°	6,0°	5,0°
4680	10,0°	11,0°	22,0°	35,0°	33,0°	28,0°	28,0°	25,0°	22,0°	18,0°	17,0°	15,0°	13,5°	11,0°	10,0°	7,0°	6,0°	5,0°
4300	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	28,0°	26,0°	23,0°	18,0°	16,0°	13,0°	11,0°	10,0°	8,5°	7,0°	6,0°	5,0°
3920	10,0°	11,0°	22,0°	35,0°	33,0°	33,0°	28,0°	26,0°	24,0°	20,0°	17,0°	15,0°	11,0°	10,0°	8,5°	6,5°	5,5°	4,5°
3540	10,0°	11,0°	22,0°	35,0°	35,0°	33,0°	28,0°	26,0°	25,0°	20,0°	15,0°	13,0°	11,0°	9,0°	7,0°	6,0°	5,0°	4,0°
3160	10,0°	11,0°	24,0°	37,0°	35,0°	33,0°	29,0°	27,0°	26,0°	19,0°	15,0°	12,0°	10,0°	8,0°	5,0°	5,0°	4,0°	3,0°
2780	10,0°	11,0°	24,0°	37,0°	35,0°	33,0°	30,0°	27,0°	24,0°	18,0°	15,0°	11,0°	7,0°	6,0°	4,0°	2,0°	1,0°	0,0°
2400	10,0°	11,0°	24,0°	37,0°	35,0°	33,0°	27,0°	24,0°	20,0°	15,0°	10,0°	7,0°	5,0°	4,0°	1,0°	-0,9°	-1,9°	-2,9°
2020	10,0°	11,0°	22,0°	35,0°	33,0°	30,0°	24,0°	21,0°	17,0°	12,0°	8,0°	5,0°	2,0°	1,0°	-1,4°	-2,9°	-3,9°	-4,9°
1640	10,0°	11,0°	20,0°	35,0°	33,0°	30,0°	23,0°	21,0°	14,0°	9,0°	5,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°
1260	10,0°	11,0°	13,0°	25,0°	28,0°	26,0°	23,0°	17,0°	12,0°	8,0°	4,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°
880	10,0°	11,0°	10,0°	20,0°	20,0°	21,0°	20,0°	15,0°	10,0°	6,0°	4,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°
500	10,0°	11,0°	10,0°	10,0°	17,0°	19,0°	18,0°	13,0°	8,0°	4,0°	4,0°	2,0°	-0,9°	-1,9°	-3,9°	-3,9°	-4,4°	-4,9°

Image 14: Ignition map with 3 bar mapsensor axis

Now we can clearly see that the ignition advance shifts over the boost pressure axis. The 35 degrees advance we see in *image 13* is set for -0.68 bar boost pressure. In the 3 bar table *(image 14)* this advance point is suddenly shifted to -0.62 bar.

This of course should not happen!! Ignition advance for -0.68 bar should remain the same.
So, the entire map should be shifted to the left a bit when using a 3 bar sensor.

An example for such an ignition map is given in *image 15*. Please note that this is not the same table but is only displayed for reference reasons.



	-1,00	-0,84	-0,76	-0,69	-0,60	-0,52	-0,36	-0,20	-0,04	0,13	0,28	0,44	0,61	0,76	0,93	1,09	1,26	1,41
6200	13,0°	23,0°	34,0°	31,0°	29,0°	28,0°	25,0°	23,0°	23,0°	20,5°	17,0°	15,0°	13,0°	12,0°	12,0°	10,0°	8,7°	8,5°
5820	13,0°	23,0°	34,0°	32,0°	29,0°	27,0°	25,0°	22,0°	21,0°	19,5°	18,5°	17,5°	13,5°	10,0°	10,0°	9,0°	8,0°	7,5°
5440	13,0°	23,0°	35,0°	32,0°	30,0°	28,0°	25,0°	23,0°	21,0°	19,0°	17,0°	15,5°	13,5°	9,0°	8,0°	7,0°	6,5°	6,0°
5060	13,0°	23,0°	35,0°	33,0°	30,0°	28,0°	25,0°	23,0°	22,0°	18,0°	15,0°	14,0°	13,0°	11,0°	7,0°	6,0°	5,5°	5,0°
4680	13,0°	23,0°	35,0°	33,0°	30,0°	28,0°	25,0°	23,0°	22,0°	19,0°	15,0°	12,0°	10,5°	9,0°	6,5°	5,5°	4,8°	4,6°
4300	13,0°	23,0°	35,0°	34,0°	31,0°	29,0°	26,0°	23,0°	21,0°	19,0°	15,0°	12,0°	8,5°	7,5°	7,0°	5,5°	4,5°	4,0°
3920	13,0°	23,0°	35,0°	36,0°	31,0°	29,0°	26,0°	24,0°	22,0°	20,0°	15,0°	12,0°	9,0°	7,5°	6,5°	5,5°	4,5°	4,0°
3540	13,0°	23,0°	35,0°	36,0°	33,0°	30,0°	27,0°	25,0°	23,0°	19,0°	15,0°	12,0°	8,0°	7,5°	6,0°	5,0°	4,0°	3,5°
3160	13,0°	23,0°	35,0°	37,0°	33,0°	30,0°	27,0°	24,0°	21,0°	18,0°	14,0°	11,0°	8,0°	5,0°	5,0°	4,0°	3,5°	2,5°
2780	13,0°	23,0°	31,0°	37,0°	32,0°	28,0°	25,0°	23,0°	19,0°	16,0°	11,0°	8,0°	6,0°	4,0°	2,0°	2,0°	1,5°	1,0°
2400	13,0°	20,0°	28,0°	35,0°	32,0°	29,0°	26,0°	21,0°	18,0°	12,0°	9,0°	6,0°	3,0°	1,0°	-0,9°	-0,9°	-0,4°	-0,9°
2020	13,0°	19,0°	25,0°	34,0°	30,0°	27,0°	22,0°	17,0°	14,0°	11,0°	6,0°	3,0°	0,0°	-1,9°	-2,9°	-2,9°	-2,4°	-2,9°
1640	13,0°	13,0°	22,0°	34,0°	28,0°	25,0°	21,0°	16,0°	12,0°	7,0°	4,0°	0,0°	-1,9°	-1,9°	-2,9°	-2,9°	-2,4°	-2,9°
1260	13,0°	13,0°	15,0°	27,0°	28,0°	24,0°	20,0°	15,0°	11,0°	7,0°	4,0°	0,0°	-1,9°	-1,9°	-2,9°	-2,9°	-2,4°	-2,9°
880	11,0°	11,0°	11,0°	19,0°	22,0°	20,0°	16,0°	10,0°	6,0°	5,0°	3,0°	-0,9°	-1,9°	-1,9°	-2,9°	-2,9°	-2,4°	-2,9°
500	10,0°	10,0°	10,0°	17,0°	19,0°	18,0°	13,0°	8,0°	4,0°	4,0°	2,0°	-0,9°	-1,9°	-1,9°	-2,9°	-2,9°	-2,4°	-2,9°

Image 15: An altered table for 3 bar sensor

Now you can see that not only the pressure maps should be altered but also every single table that has a boost pressure dependency. One can easily understand the complexity of the conversion problem. In the next chapters a summary will be given for known tables that have this dependency.

Maps that should be considered when converting to 3 bar sensor

Symbolname	Description	Dependency
Tryck_mat	Boost table for manual transmission	Values are boost pressure
Tryck_mat_a	Boost table for automatic transmission	Values are boost pressure
Tryck_vakt_tab	Boost limiter	Values are boost pressure
Regl_tryck_fgaut	Boost limiter 1st gear automatic	Values are boost pressure
Regl_tryck_fgm	Boost limiter 1st gear manual	Values are boost pressure
Regl_tryck_sgm	Boost limiter 2nd gear manual	Values are boost pressure
Limp_tryck_konst	Limp home mode maximum boost	Values are boost pressure
Idle_tryck	Boost level on idle	Values are boost pressure
Insp_mat	Fuel injection correction table	Boost pressure in x axis
Ign_map_0	Ignition map for warm engine (main map)	Boost pressure in x axis
Ign_map_2	Ignition map for knock detection	Boost pressure in x axis
Ign_map_3	Ignition map transient	Boost pressure in x axis
Ign_map_4	Ignition map for cool engine (warmup map)	Boost pressure in x axis
Ign_map_6	Ignition map for torque reduction upshift	Boost pressure in x axis
Ign_map_7	Ignition map for torque reduction downshift	Boost pressure in x axis
Ign_map_8	Ignition map for idle during speed	Boost pressure in x axis
Lambdamatris	Lambda sensor control factors	Boost pressure in y axis
Knock_press_tab	Pressure limit for knock indication	Values are boost pressure
Purge_tab	Purge matrix (0 – 100%)	Boost pressure in x axis
Del_mat	Delay map (crankshaft degrees)	Boost pressure in x axis
Temp_reduce_mat	Fuel Injection, Temperature compensation reduction map (open throttle)	Boost pressure in x axis
Temp_reduce_mat_2	Fuel Injection, Temperature compensation reduction map (closed throttle)	Boost pressure in x axis
Mis1000_map	Misfire maximum allowed (emission level)	Boost pressure in x axis
Mis200_map	Misfire maximum allowed (cat overheating)	Boost pressure in x axis
Misfire_map	⚠ T5.2 only	Boost pressure in x axis
Detect_map	Reference map for misfire detection	Boost pressure in x axis
Knock_ref_matrix	Knock reference map	Boost pressure in x axis
Turbo_knock_tab	Pressure limit for turbo knock regulation	Values are boost pressure
Iv_min_load	Pressure for which adaption should be adjusted	Values are boost pressure
Min_load_gadapt	Minimum load for global adaption	Values are boost pressure
Max_load_gadapt	Maximum load for global adaption	Values are boost pressure
Kadapt_load_low	Pointed adaption minimum load	Values are boost pressure
Kadapt_load_high	Pointed adaption maximum load	Values are boost pressure
Last_cyl_komp		Values are boost pressure
Open_loop	Load limit where system switches to openloop	Values are boost pressure
Open_loop_adapt	Open loop adaption (not used?)	Values are boost pressure
Open_loop_knock	Load limit where system switches to open loop during knock conditions.	Values are boost pressure
Lacc_clear_tab		Values are boost pressure
Lner_detekt		Values are boost pressure
Lupp_detekt		Values are boost pressure
Sond_heat_tab	Load limit for lambda sonde heating	Values are boost pressure
Grund_last		Values are boost pressure
Grund_last_max		Values are boost pressure

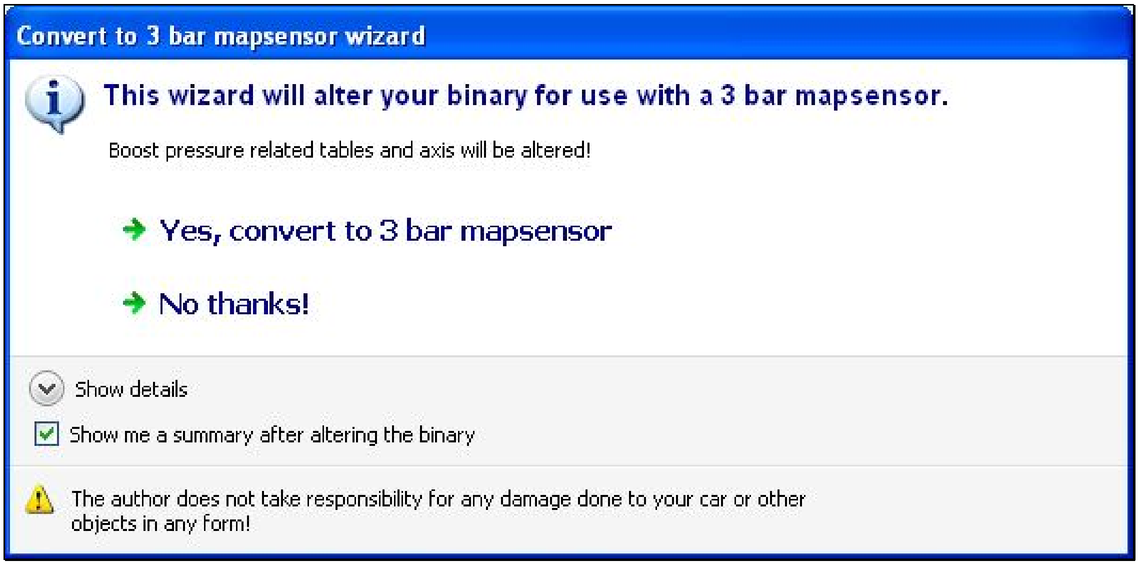
Still investigating tables:

Overs_tab_xaxis

Implementation in T5Suite

T5Suite has a "convert to 3 bar mapsensor "wizard" build in that allows you to alter your binary so that all the before mentioned parameters are automatically changed.

NOTE: Converting to a 3 bar mapsensor is a very complex procedure. You will have to verify all settings after converting the binary and check operation using knock sensing LED, EGT meter and wideband lambda sensor.



Convert to 3 bar mapsensor wizard

This wizard will alter your binary for use with a 3 bar mapsensor.

Boost pressure related tables and axis will be altered!

☑ Show me a summary after altering the binary

The author does not take responsibility for any damage done to your car or other objects in any form!

Converting T5 to larger injectors

This chapter will explain the issues that one will encounter when converting to larger injectors. The standard injectors deliver 345 cc of fuel @ 3 bar pressure.
The approximate maximum power achievable with these injectors is ~300 bhp. When a car is to be tuned over this power, larger injectors are needed to be able to supply the amount of fuel needed for this.

Injector information

Stock injectors on Saab 9000 1993

Brand: Lucas
Mfd. p/n:D5162EA
Imp: 15.9 Ohms
Flow: 286g C7H16 @ 3 bar / 60s (~408 cc/min)

Stock injectors on Saab 9000 1994-1998

345 cc/min @ 3 bar fuel pressure

Static Flow Rate: 31.6 lb/hr @ 43.5PSI = 239.4 g/min = 349 cc/min (+/-4%)
Dynamic Flow Rate 2.5ms pulse width @ 100Hz: 6.68 mg/pulse (+/-6%)
Coil Resistance: 16.2 Ohms (+/-0.35 Ohms)
Physical Dimensions: EV1 Body Type
Connector: Minitimer (Bosch EV6)

Possible upgrades can be:

Bosch -968 "Green Giants"

465 cc/min @ 3.8 bar fuel pressure.
413 cc/min @ 3.0 bar fuel pressure.

Siemens Deka 630 cc/min @ 3.0 bar fuel pressure.

Static Flow Rate (GAS): 62.7 lb/hr = 646 cc/min = 474 g/min
Dynamic Flow Rate 2.5ms PW @ 100Hz (GAS): 20.2mg/pulse
Coil Resistance: 12 Ohms

Gain: 0.11ms/mg
Offset: 0.055ms
Turn on time @ 14VDC: 1.14ms
Turn off time: 0.85ms @ 600KPa
Factory Tolerance: +/-6 %
Spray Pattern: Cone (4-Hole)
Connector: Minitimer (Bosch EV1)

Siemens Deka 4 875 cc/min @ 3.0 bar fuel pressure.

Static Flow Rate (GAS): 80 lb/hr = 875 cc/min
Dynamic Flow Rate 2.5ms PW @ 100Hz (GAS): 18.6 mg/pulse
Coil Resistance: 12 Ohms
Spray Pattern: Pencil
Connector: Minitimer

Bosch EV14 920 cc/min @ 3.0 bar fuel pressure.

Static Flow Rate (GAS): 88 lb/hr = 920 cc/min
Coil Resistance: 12 Ohms

Technical information

Injectors used in Saabs with Trionic 5 engine management systems are of the high impedance type. This means the impedance (resistance) is ~12 Ω. The other type – low impedance - is not used in conjunction with T5. The low impedance type are ~3 Ω.

Injectors are actually little solenoid valves which open when current flows through them. By pulsing the injectors the right amount of times per interval the valves are opened and closed in the same frequency as engine RPM. This way the fuel can be injected at precisely the right moment.

The waveform supplied to a single injector could look something like in *image 16*.

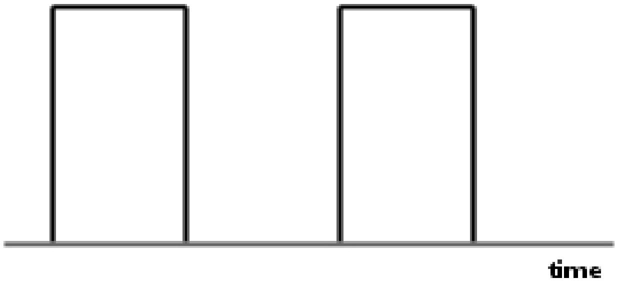


Image 16: pulsing of injector

Theoretically the injector would open at the point where the voltage goes high and close where the voltage goes low.

By changing the width of the pulse the effective opening time per period can be changed and thus the amount of fuel injected per cycle. When less fuel is needed as shown in *image 16* the width is lowered and could result in *image 17*.

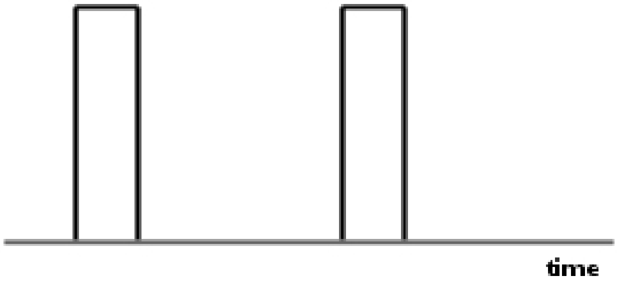


Image 17: Smaller pulses result in less fuel

When more fuel is needed the waveform will get wider and could look something like *image 18*. This is called Pulse Width Modulation (PWM).

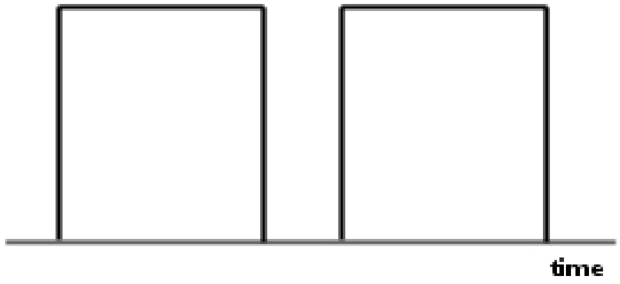


Image 18: Wider pulses result in more fuel

In these examples the frequency stays the same (RPM) while the pulse width changes (throttle opening and therefore fuel requirement). When engine RPM rises the amount of pulses per period of time would simply increase. This is shown in *image 19*.

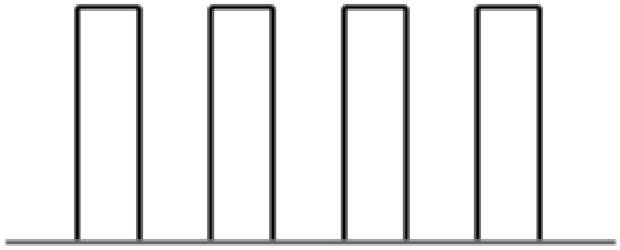


Image 19: Frequency increased with higher RPM

This image instantly clarifies why injectors are limited to a certain level of power. The maximum opening time that can be reached is determined by the maximum rpm value the engine can achieve. When engine speed rises the duration that the fuel can be injected decreases. At the point of maximum engine speed this duration will be at its minimum. The effective injection duration in other rpm sites cannot be higher than at this point.
The PWM duty cycle is the percentage between on and off time. A typical maximal duty cycle is ~80%. This means the injectors flow rate * 0.8 is about the maximum amount of fuel the injector can deliver in action.

Converting math

When calculating fuel flow rate from one rail pressure to another we'll have to use this formula:

$\mathbf{R_n = \sqrt{(P_n / P_o)} * R_o}$

R_n = New flow rate for the new rail pressure.
P_n = New pressure on the rail.
P_o = Old pressure on the rail.
R_o = Old flow rate at the old rail pressure.

For example we need to recalculate the flow rate for the green giant injectors for 3 bar rail pressure. The rated flow @ 3.8 bar is given by the manufacturer and is 465 cc / minute.
The formula will be:

Rn = √(3.0 / 3.8) * 465 = ~ 413 cc / minute

When we want to calculate the rail pressure to get a certain flow rate from our injectors we'll have to use this formula:

$\mathbf{P_n = P_o * (R_n / R_o)^2}$

P_n = New pressure that needs to be applied
P_o = Old pressure (current)
R_n = Desired rate of flow for injector
R_o = Old flow (current)

For example we want to get 370 cc of flow from our stock injectors.
The new pressure would then be:
3.0 * (370/345)^2 = ~ 3.45 bar

Real world

In reality injectors don't open instantly. They need time to open the valve. Dramatized a bit the actual opening time will look something like the red line in *image 20*. Opening and closing ramps must be taken into consideration. In an ideal world the injector would go from closed to open on the exact moment the pulse is applied. In high resistance injector injection systems, like the T5, this is a factor that must be taken into consideration.

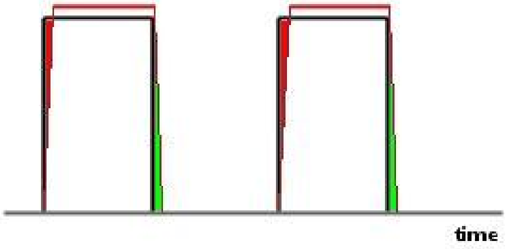


Image 20: Pulse vs. injector open time

What is apparent is that fuel starts to flow at the start of the pulse but the injector's flow rate isn't reached until a certain time after the pulse was applied. At the end of the pulse we must take the closing time into account. Note that the opening and closing ramps are asymmetrical, this means that the ramp faults do not cancel each other out. The fuel we are missing is denoted by the red colour and the extra fuel is denoted by the green colour. There can also be differences in the height of the figures due the differences between injectors. Bosch standard injectors have small manufacturing tolerances so the differences are thereby small. When dealing with larger injectors, even small differences in percentage leads to large real differences. In this case we must therefore consider flow matched injectors. The Siemens 630 cc has a reputation of having large differences in flow rate, especially when measuring on low and mid range duty cycle. Differences in flow rate could lead to pre ignition/detonation when one cylinder is running lean and the others are running rich because the injectors inject different amount of fuel. The knock regulation in Trionic 5 should protect the engine from failures but it can result in rough idle and a roller coaster like wideband lambda readout on idle and mid-range RPM.
The opening time is in direct relation to the battery voltage and the relative fuel pressure, the closing time is in direct relation to the relative fuel pressure. Therefore one idea could be to adjust the Batt_korr_tab! to cancel out the difference in opening time when converting to larger injectors. The Batt_korr_tab! table regulates the time the injectors are held open in respect of the battery voltage because of the relationship of how fast the injectors open when different voltages are applied.

The most important thing to remember is that larger injectors need more time to open than the stock injectors. The fuel flow maybe better (more fuel per microsecond passes the valve simply because the opening is larger) but the valve is a bit bigger too and hence needs more time to open. On low rpm values this might not be a very big issue but on higher rpm ranges the frequency of the waveform is also higher and hence the injectors response time becomes more important (there's simply less time per cycle to get the fuel into the intake manifold). If we replace the injectors from *image 20* to larger ones the new opening times could look something like *image 21*.

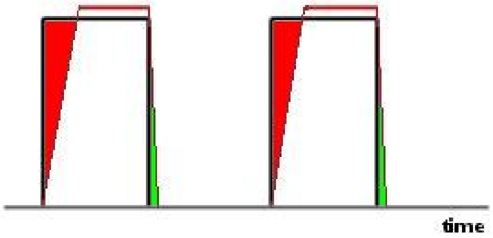
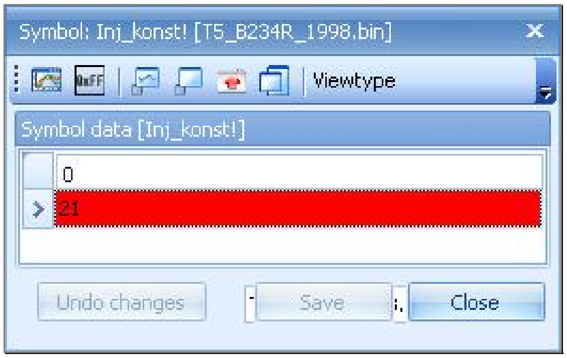


Image 21: Pulse vs. injector open time with larger injectors

These two images make the concrete problem very visible. The opening duration multiplied by the flow rate determines the effective amount of fuel injected. Larger injectors need a little more time to open – hence the actual opening time gets smaller – but the amount of fuel per microsecond is larger. Setting up the software in such a way that the injectors deliver the required amount of fuel at all RPM and load ranges takes some work.

Simple conversion

Up to ~25% larger injectors can be used without many problems. The change in opening times are relative small and can be altered using a constant in the software. The remainder of the difference in injected fuel can be adapted by the Trionic using the lambda values measured (there's a limit to this adaption of course).


Symbol: Inj_konst! [T5_B234R_1998.bin]

Symbol data [Inj_konst!]

0
21

Image 22: Injector constant shown in T5Suite

If we look at the Trionic software we find that there's a constant called Inj_konst present. This constant lets us influence the factor used in the fuel injection calculation. In stock software this constant is mostly set to ~20.
When we upgrade to larger injectors we need to decrease this constant (opening time needs to be smaller with larger injectors) by the same percentage that the injectors were increased.

Examples

Fuel pressure upgrade

We upgrade the fuel pressure from 3.0 (44 psi) bar to 3.8 bar (55 psi).
If we use the math on this pressure rise we'll find out that the stock injectors will deliver:

$R_n = \sqrt{(3.8 / 3.0)} * 345 = 388$ cc/min

The factor of extra fuel will be 388/345 = 1.125

The Inj_konst value in the software used was 21. Now we divide this value with 1.125 and this results in ~19. Since we must round the figure to an integer value we might need to fine-tune the injection table (insp_mat) to make sure we don't have any lean/rich spots using a wideband lambda sensor.

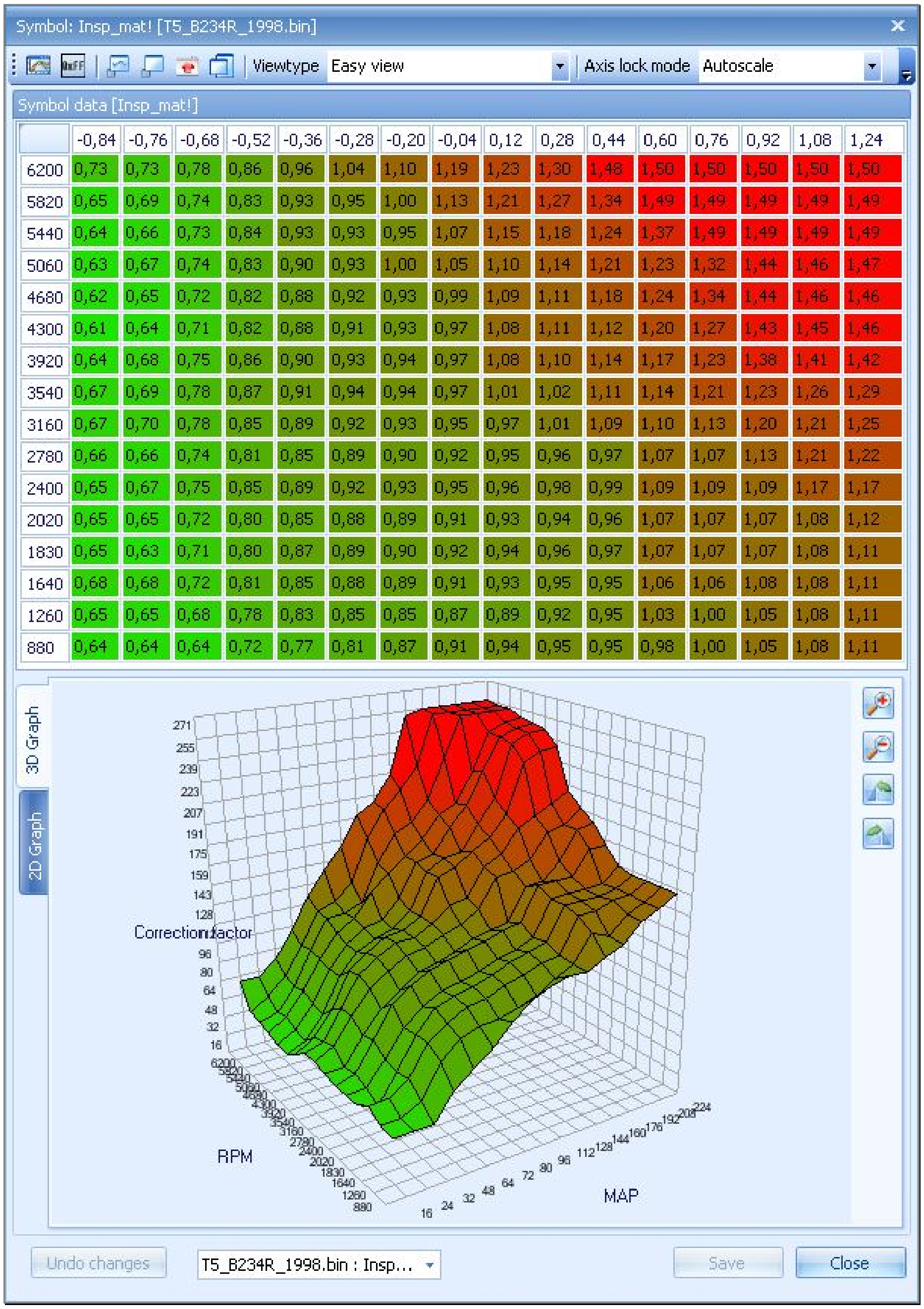
Injector upgrade

We upgrade our stock injectors with larger Bosch "green giant" injectors.
We leave the fuel pressure @ 3 bar.
The larger injectors will deliver 465 cc of fuel @ 3.8 bar.
At 3 bar these injectors will deliver ~413 cc of fuel per minute.
Our gain factor in this case will be 413/345 = 1.197.
If we recalculate the injector constant (21 / 1.197) = 17.5, we'll have to enter 18.
Again we have to round the result and we might need fine-tuning of the injection map (insp_mat) to make sure we don't have any lean/rich spots using a wideband lambda sensor.

Advanced conversion

If we want to replace our stock injectors with much larger injectors we'll have to do more than only altering the injector constant.
Trionic won't be able to adapt the injection time difference that will occur because of the injector opening latency when the difference in injector flow is more than ~25%.

In this case we'll need to alter the injector constant as well as remap the entire injection map (insp_mat).



Symbol: Insp_mat! [T5_B234R_1998.bin]

Viewtype: Easy view

Axis lock mode: Autoscale

Symbol data [Insp_mat!]

	-0,84	-0,76	-0,68	-0,52	-0,36	-0,28	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24
6200	0,73	0,73	0,78	0,86	0,96	1,04	1,10	1,19	1,23	1,30	1,48	1,50	1,50	1,50	1,50	1,50
5820	0,65	0,69	0,74	0,83	0,93	0,95	1,00	1,13	1,21	1,27	1,34	1,49	1,49	1,49	1,49	1,49
5440	0,64	0,66	0,73	0,84	0,93	0,93	0,95	1,07	1,15	1,18	1,24	1,37	1,49	1,49	1,49	1,49
5060	0,63	0,67	0,74	0,83	0,90	0,93	1,00	1,05	1,10	1,14	1,21	1,23	1,32	1,44	1,46	1,47
4680	0,62	0,65	0,72	0,82	0,88	0,92	0,93	0,99	1,09	1,11	1,18	1,24	1,34	1,44	1,46	1,46
4300	0,61	0,64	0,71	0,82	0,88	0,91	0,93	0,97	1,08	1,11	1,12	1,20	1,27	1,43	1,45	1,46
3920	0,64	0,68	0,75	0,86	0,90	0,93	0,94	0,97	1,08	1,10	1,14	1,17	1,23	1,38	1,41	1,42
3540	0,67	0,69	0,78	0,87	0,91	0,94	0,94	0,97	1,01	1,02	1,11	1,14	1,21	1,23	1,26	1,29
3160	0,67	0,70	0,78	0,85	0,89	0,92	0,93	0,95	0,97	1,01	1,09	1,10	1,13	1,20	1,21	1,25
2780	0,66	0,66	0,74	0,81	0,85	0,89	0,90	0,92	0,95	0,96	0,97	1,07	1,07	1,13	1,21	1,22
2400	0,65	0,67	0,75	0,85	0,89	0,92	0,93	0,95	0,96	0,98	0,99	1,09	1,09	1,09	1,17	1,17
2020	0,65	0,65	0,72	0,80	0,85	0,88	0,89	0,91	0,93	0,94	0,96	1,07	1,07	1,07	1,08	1,12
1830	0,65	0,63	0,71	0,80	0,87	0,89	0,90	0,92	0,94	0,96	0,97	1,07	1,07	1,07	1,08	1,11
1640	0,68	0,68	0,72	0,81	0,85	0,88	0,89	0,91	0,93	0,95	0,95	1,06	1,06	1,08	1,08	1,11
1260	0,65	0,65	0,68	0,78	0,83	0,85	0,85	0,87	0,89	0,92	0,95	1,03	1,00	1,05	1,08	1,11
880	0,64	0,64	0,64	0,72	0,77	0,81	0,87	0,91	0,94	0,95	0,95	0,98	1,00	1,05	1,08	1,11

Image 23: Injection map shown in T5Suite

Since the injection map is already mapped up to its limit in the stock software we need to make sure that we alter the injector constant in such a way that too much fuel is injected. That way, we can drop the injection map later on to get the fueling spot on. If we would change the injector constant to a lower value we might have to increase the injection map which is no longer possible because it is already maxed out in high rpm/high load.

Suppose we get the Siemens injectors instead of our stock injectors. The injectors will deliver 630 cc of fuel per minute. This is 630/345 = 1.83 times more than stock.

We'll have to adjust the injector constant with this factor also. Because we want to be able to alter the injection map afterwards (and we cannot put higher values in it so we'll have to get lower) we have to make sure that the injector constant is a bit higher than we calculate.

The new injector constant would be around (21 / 1.83) = 11.4
Because we need to be able to alter the injection map afterwards we would have to enter 12 or 13 for the injector constant.

Now remember the extra time the injector needs to be opened. Because there's no standard way of calculating this we need to increase the injector constant from the value we have up to the point where the engine starts to run rich. I've seen 630cc injectors used with an injector constant of 17 (!).

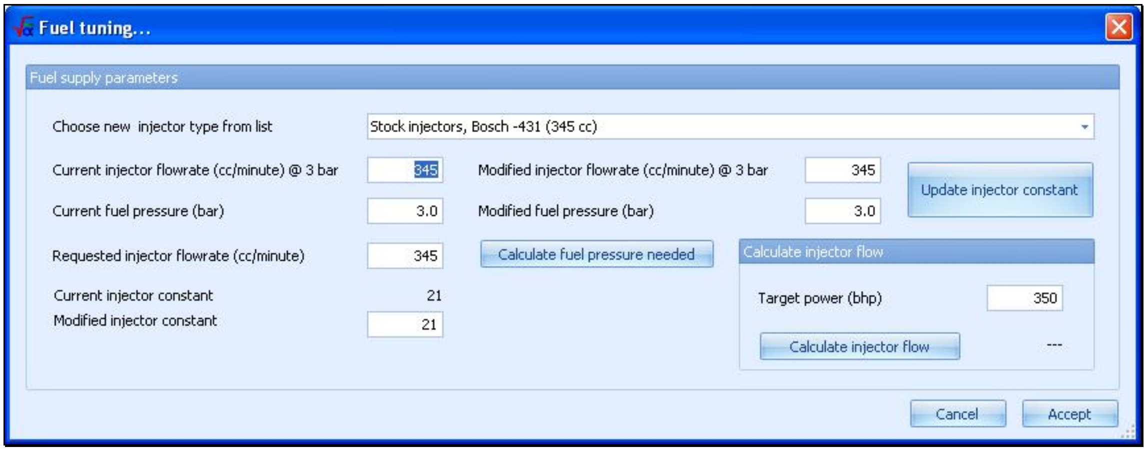
Now we need to remap the injection matrix site by site. We'll see that we need to decrease the values in the map because the engine was already running rich.
To be able to remap the injection matrix properly you'd ideally need an exhaust gas temperature meter (EGT), a wide band lambda sensor + display and a rolling road.
The latter is to make sure you can determine on which load and rpm site you are working.

Implementation in T5Suite

T5Suite gives the user the possibility to re-calculate the injector constant with given injector flowrates and fuel pressure values.
The above formulas have been implemented for this with one addition.
To compensate for the injector latency increase in larger injectors (more time needed to open) T5Suite increases the newly calculated injector constant with one (1) for every 100 cc / minute increase in injector flow @ 3 bar. This is only guesstimating but there's nothing better at this time.

Again: when injectors flow changes by more than 25 percent the injector constant and injection map needs verification too.

NOTE: minimum time for normal idle ~ 1.2 ms.



Choose new injector type from list: Stock injectors, Bosch -431 (345 cc)

Current injector flowrate (cc/minute) @ 3 bar: 345

Modified injector flowrate (cc/minute) @ 3 bar: 345

Current fuel pressure (bar): 3.0

Modified fuel pressure (bar): 3.0

Requested injector flowrate (cc/minute): 345

Current injector constant: 21

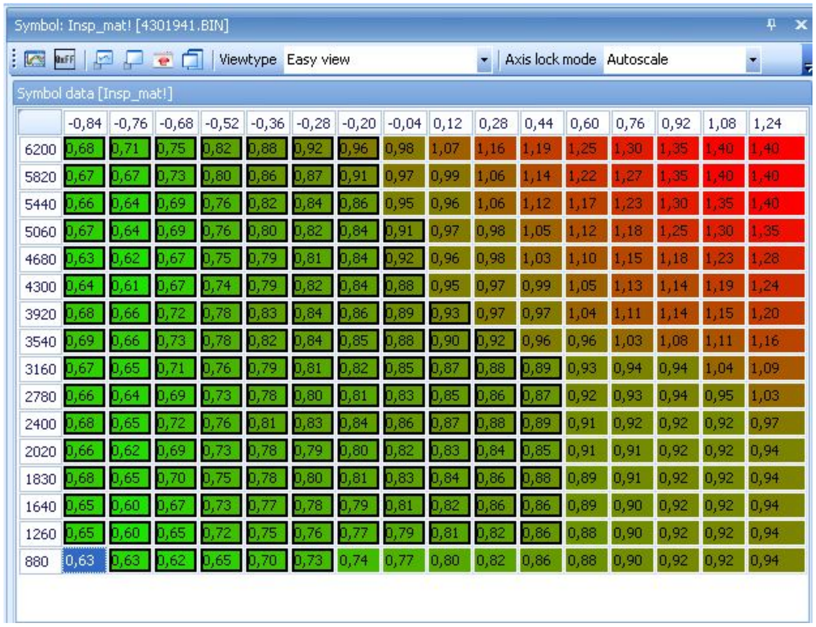
Modified injector constant: 21

Target power (bhp): 350

Adjusting supporting axis

If you want to rescale maps – for example – to support higher engine speeds you need to perform a couple of steps.

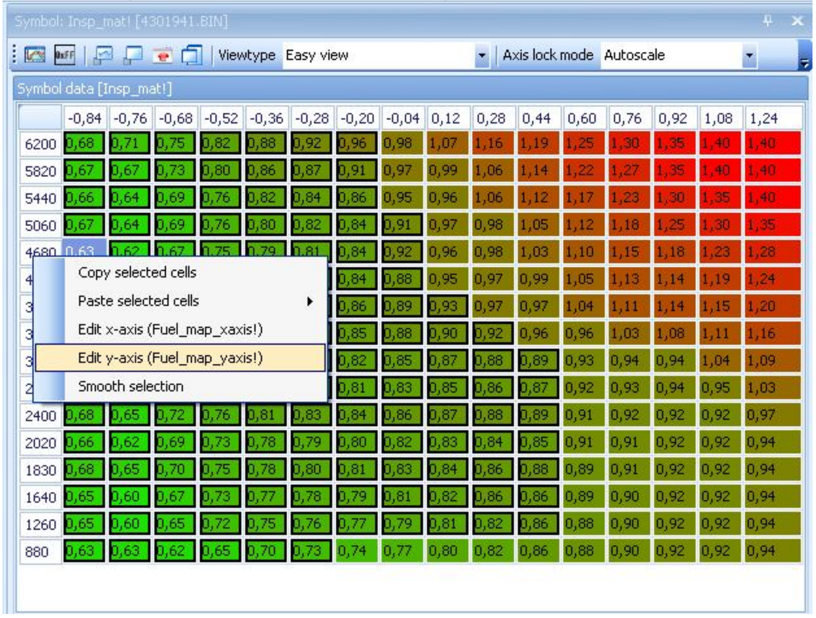
Open the map you want the axis changed for.



Viewtype: Easy view
Axis lock mode: Autoscale

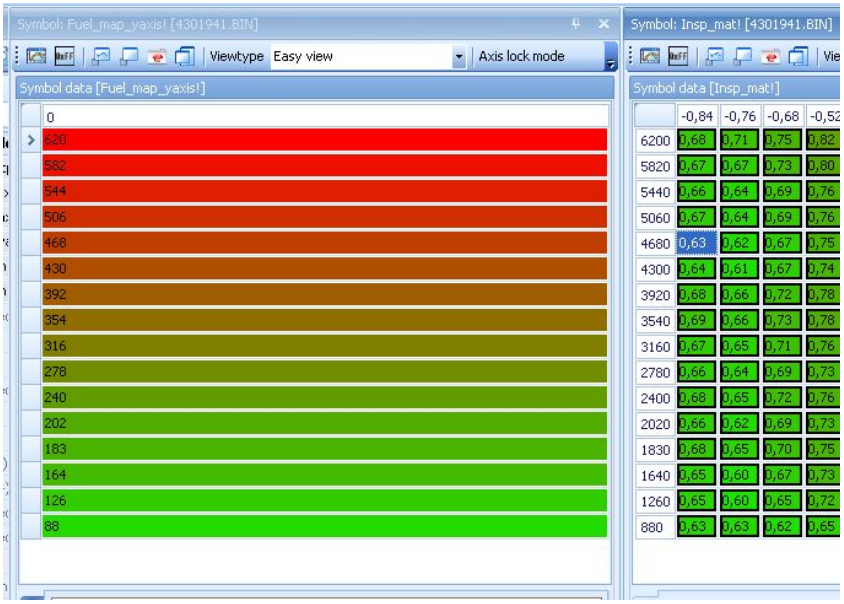
	-0,84	-0,76	-0,68	-0,52	-0,36	-0,28	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24
6200	0,68	0,71	0,75	0,82	0,88	0,92	0,96	0,98	1,07	1,16	1,19	1,25	1,30	1,35	1,40	1,40
5820	0,67	0,67	0,73	0,80	0,86	0,87	0,91	0,97	0,99	1,06	1,14	1,22	1,27	1,35	1,40	1,40
5440	0,66	0,64	0,69	0,76	0,82	0,84	0,86	0,95	0,96	1,06	1,12	1,17	1,23	1,30	1,35	1,40
5060	0,67	0,64	0,69	0,76	0,80	0,82	0,84	0,91	0,97	0,98	1,05	1,12	1,18	1,25	1,30	1,35
4680	0,63	0,62	0,67	0,75	0,79	0,81	0,84	0,92	0,96	0,98	1,03	1,10	1,15	1,18	1,23	1,28
4300	0,64	0,61	0,67	0,74	0,79	0,82	0,84	0,88	0,95	0,97	0,99	1,05	1,13	1,14	1,19	1,24
3920	0,68	0,66	0,72	0,78	0,83	0,84	0,86	0,89	0,93	0,97	0,97	1,04	1,11	1,14	1,15	1,20
3540	0,69	0,66	0,73	0,78	0,82	0,84	0,85	0,88	0,90	0,92	0,96	0,96	1,03	1,08	1,11	1,16
3160	0,67	0,65	0,71	0,76	0,79	0,81	0,82	0,85	0,87	0,88	0,89	0,93	0,94	0,94	1,04	1,09
2780	0,66	0,64	0,69	0,73	0,78	0,80	0,81	0,83	0,85	0,86	0,87	0,92	0,93	0,94	0,95	1,03
2400	0,68	0,65	0,72	0,76	0,81	0,83	0,84	0,86	0,87	0,88	0,89	0,91	0,92	0,92	0,92	0,97
2020	0,66	0,62	0,69	0,73	0,78	0,79	0,80	0,82	0,83	0,84	0,85	0,91	0,91	0,92	0,92	0,94
1830	0,68	0,65	0,70	0,75	0,78	0,80	0,81	0,83	0,84	0,86	0,88	0,89	0,91	0,92	0,92	0,94
1640	0,65	0,60	0,67	0,73	0,77	0,78	0,79	0,81	0,82	0,86	0,86	0,89	0,90	0,92	0,92	0,94
1260	0,65	0,60	0,65	0,72	0,75	0,76	0,77	0,79	0,81	0,82	0,86	0,88	0,90	0,92	0,92	0,94
880	0,63	0,63	0,62	0,65	0,70	0,73	0,74	0,77	0,80	0,82	0,86	0,88	0,90	0,92	0,92	0,94

Suppose we want to change the Y-axis (RPM) here. Right click on the Y axis and select "Edit y-axis)



	-0,84	-0,76	-0,68	-0,52	-0,36	-0,28	-0,20	-0,04	0,12	0,28	0,44	0,60	0,76	0,92	1,08	1,24
6200	0,68	0,71	0,75	0,82	0,88	0,92	0,96	0,98	1,07	1,16	1,19	1,25	1,30	1,35	1,40	1,40
5820	0,67	0,67	0,73	0,80	0,86	0,87	0,91	0,97	0,99	1,06	1,14	1,22	1,27	1,35	1,40	1,40
5440	0,66	0,64	0,69	0,76	0,82	0,84	0,86	0,95	0,96	1,06	1,12	1,17	1,23	1,30	1,35	1,40
5060	0,67	0,64	0,69	0,76	0,80	0,82	0,84	0,91	0,97	0,98	1,05	1,12	1,18	1,25	1,30	1,35
4680	0,63	0,62	0,67	0,75	0,79	0,81	0,84	0,92	0,96	0,98	1,03	1,10	1,15	1,18	1,23	1,28
4							0,84	0,88	0,95	0,97	0,99	1,05	1,13	1,14	1,19	1,24
3							0,86	0,89	0,93	0,97	0,97	1,04	1,11	1,14	1,15	1,20
3							0,85	0,88	0,90	0,92	0,96	0,96	1,03	1,08	1,11	1,16
3							0,82	0,85	0,87	0,88	0,89	0,93	0,94	0,94	1,04	1,09
2							0,81	0,83	0,85	0,86	0,87	0,92	0,93	0,94	0,95	1,03
2400	0,68	0,65	0,72	0,76	0,81	0,83	0,84	0,86	0,87	0,88	0,89	0,91	0,92	0,92	0,92	0,97
2020	0,66	0,62	0,69	0,73	0,78	0,79	0,80	0,82	0,83	0,84	0,85	0,91	0,91	0,92	0,92	0,94
1830	0,68	0,65	0,70	0,75	0,78	0,80	0,81	0,83	0,84	0,86	0,88	0,89	0,91	0,92	0,92	0,94
1640	0,65	0,60	0,67	0,73	0,77	0,78	0,79	0,81	0,82	0,86	0,86	0,89	0,90	0,92	0,92	0,94
1260	0,65	0,60	0,65	0,72	0,75	0,76	0,77	0,79	0,81	0,82	0,86	0,88	0,90	0,92	0,92	0,94
880	0,63	0,63	0,62	0,65	0,70	0,73	0,74	0,77	0,80	0,82	0,86	0,88	0,90	0,92	0,92	0,94

A new map will appear called "Fuel_map_yaxis!". This map contains the axis numbers for the given map.


Symbol: Fuel_map_yaxis! [4301941.BIN]
Viewtype
Axis lock mode
Symbol: Insp_mat! [4301941.BIN]
Vie

Viewtype: Easy view

Symbol data [Fuel_map_yaxis!]

0
620
582
544
506
468
430
392
354
316
278
240
202
183
164
126
88

Symbol data [Insp_mat!]

	-0,84	-0,76	-0,68	-0,52
6200	0,68	0,71	0,75	0,82
5820	0,67	0,67	0,73	0,80
5440	0,66	0,64	0,69	0,76
5060	0,67	0,64	0,69	0,76
4680	0,63	0,62	0,67	0,75
4300	0,64	0,61	0,67	0,74
3920	0,68	0,66	0,72	0,78
3540	0,69	0,66	0,73	0,78
3160	0,67	0,65	0,71	0,76
2780	0,66	0,64	0,69	0,73
2400	0,68	0,65	0,72	0,76
2020	0,66	0,62	0,69	0,73
1830	0,68	0,65	0,70	0,75
1640	0,65	0,60	0,67	0,73
1260	0,65	0,60	0,65	0,72
880	0,63	0,63	0,62	0,65

Now you can edit this map with axis values, save it and reload the original map (insp_mat in this case). The map will now have the new axis values.

WARNING

NOTE: Please note that changing the axis values WILL cause Trionic to lookup different cells from a map and so, you will have to rescale the maps content as well (in this case, insp_mat). If you don't rescale everything the way it is supposed to be Trionic will fetch the wrong values from the map while the engine is running which may result in engine damage.
NOTEII: Also note that support axis may be used by more than one map. Please check the axis browser in T5Suite to confirm that there are no more maps that you need to rescale other than the one you started out with.

CAN Bus interface

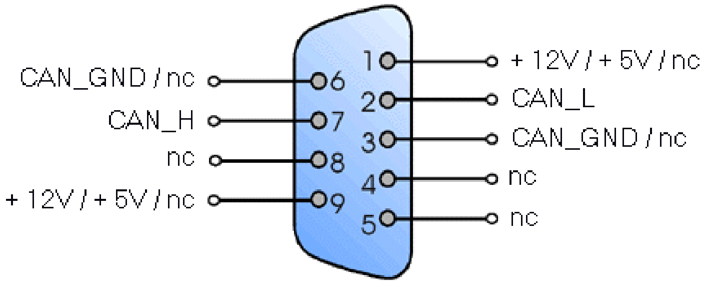
When interfacing with the Trionic T5.x unit through the CAN bus it is possible to retrieve data from the ECU. The CAN interface supplies information about runtime variables (symbols) and allows for flashing and reading the flash content (to be developed) in a faster way than the BDM interface does.

General information

Chip used on Trionic side: Intel AN825256
Communication speed used: 615 Kbit/s

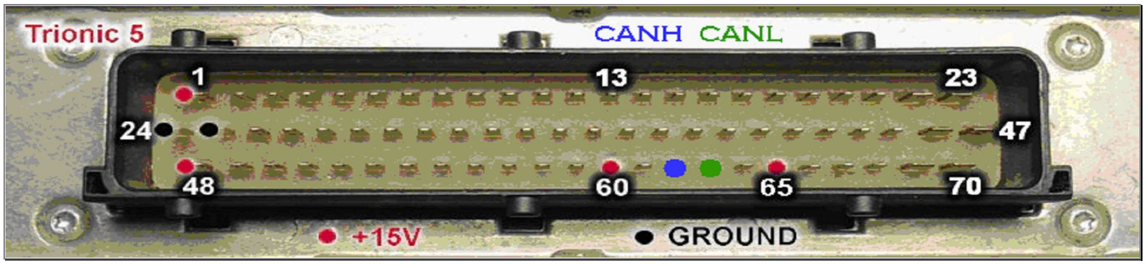
The most frequently used interface for this is the Lawicel CANUSB interface that can be found on www.canusb.com. This interface can convert CAN signals onto you USB port and vice versa. The interface has a USB port on one side – that connects to you computer – and an male RS232 (DB9) connector on the other side. This side connects to the CAN bus of the Trionic.

The Lawicel interface has the following pinout on the DB9 connector.



Connecting to CAN bus with ECU on your desk

The CAN bus is connected to the ECU through pin 62 (CANH) and pin 63 (CANL). Of course ground is also required. To connect the CAN interface directly to the Trionic, use this schema. If you want to connect to the Trionic unit in the car please check the next paragraph that handles the SFI test connector.



CAN Bus (SFI) test connector

In most **9000** T5 cars the connection for the CAN Bus is located in an SFI test connector that is located near the ECU. It is a six pin connector for which the pinning is given below.

Pinnumber	Description	Color	Trionic pin
1	CAN H	Grey	62
2	CAN L	Red	63
4	Ignition + (15)	Green/white	From fuse box
5	Ground	Black	24 & 25
6	+15 V (Programming voltage)	Red/white	65

The location of the pins in the connector housing – looking on the cable side of the connector - is shown in *image 24*. These numbers are also embossed into the plastic on the side of the connector itself.

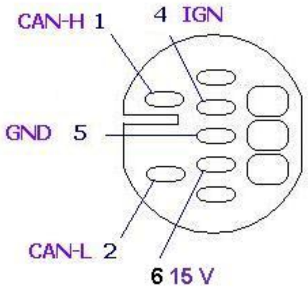
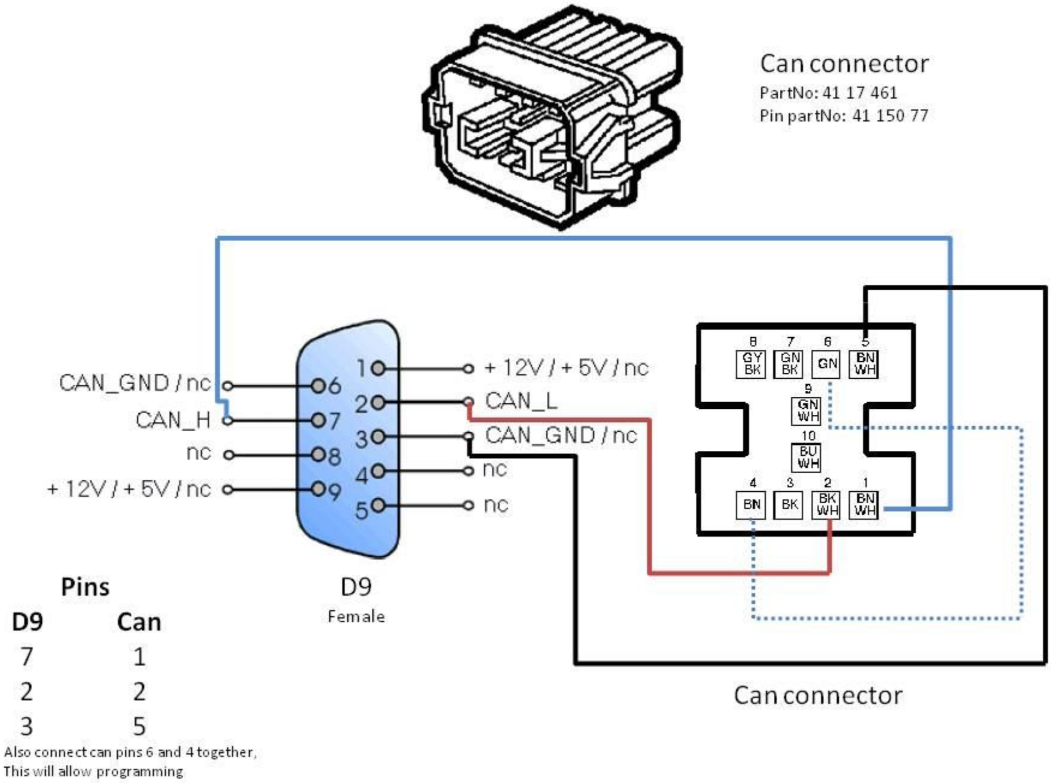


Image 24: SFI connector pinlocation

SFI Connector in-car has partnumber: 44 11 518
SFI Connector counterpart (that you need) has partnumber: 44 10 510
SFI Connector terminal (pins) used for this connection are partnumber: 12 79 06 14 from which you will need at least 3 (CAN-H, CAN-L and GND).

In **9^3 and 900NG** cars the connector is different and looks like the image shown below.

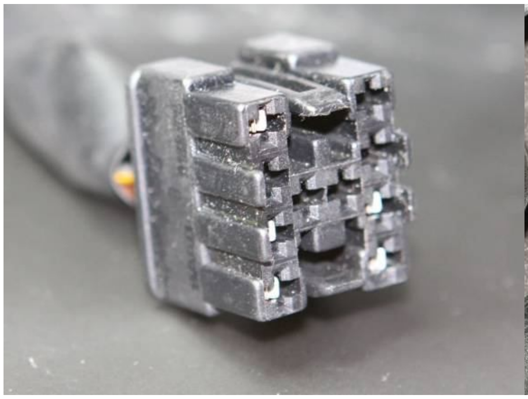


Pins

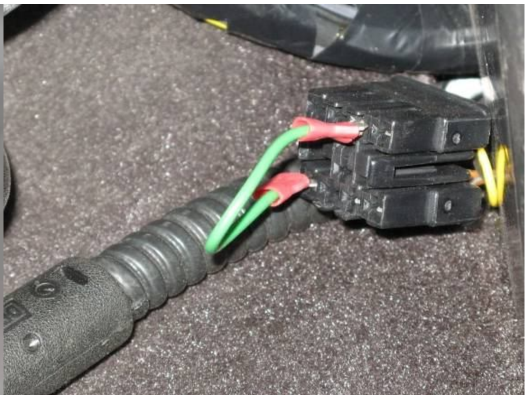
D9	Can
7	1
2	2
3	5

Also connect can pins 6 and 4 together,
This will allow programming

900NG connector



9^3 connector



CAN Bus termination

A CAN bus interface needs to be terminated by 120 Ω on both sides. This means a resistor should be soldered over CANH and CANL lines as close as possible on either side of the line.
But… Trionic is already terminated internally by 120 Ω. This is probably done because the CAN interface on Trionic would never be used as a real bus architecture, it would always be a one-to-one connection. This voids the need for adding a resistor on the Trionic side.
For the best result the cable used should be twisted (CANH and CANL) up to approx. 1 cm from the connectors.

CAN Bus protocol

SAAB engineers have implemented a proprietary protocol on top of the CAN Bus layer. The CAN layer is the transport layer in the OSI model while the SAAB protocol is the application layer.
This chapter will describe the message structure used in transferring data to and from Trionic over the CAN Bus. The protocol for sending and receiving data to and from SRAM is relatively easy. All CAN messages sent to Trionic have CAN message ID 0xC4 while messages received from Trionic are message ID 0xC6.

Low level commands

Send command byte for read

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	0x00	0x00	0x00	CMD	0xC4

Send command byte

Byte	0	1	2	3	4	5	6	7
Data	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	CMD	0xC4

Send acknowledge

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xC6

Set address for uploading bootloader

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	LENGTH	ADR_LL	ADR_LH	ADR_HL	ADR_HH	0xA5

After set address: #frames to write data into SRAM

Byte	0	1	2	3	4	5	6	7
Data	BYTE7	BYTE6	BYTE5	BYTE4	BYTE3	BYTE2	BYTE1	INDEX

Note: INDEX = 0x00 – 0x7F

Jump to address (after uploading bootloader, to start executing the bootloader)

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	ADR_LL	ADR_LH	ADR_HL	ADR_HH	0xC1

Get flash end address

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xC3
Response	0xB8	0x89	0xFF	0xFF	0x07	0x00	0x00	0xC3

Set Analogue to digital register values

Byte	0	1	2	3	4	5	6	7
Data	STATUS	AD3	AD2	AD1	AD0	ID	BANK0	0xC5
Data	???	AD7	AD6	AD5	AD4	ID	BANK1	0xC5

Note: BANK = 0x00 or 0x01

Read data from SRAM

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	ADR_LL	ADR_LH	ADR_HL	ADR_HH	0xC7
Response	(ADR+5)	(ADR+4)	(ADR+3)	(ADR+2)	(ADR+1)	B(ADR)	0x00	0xC7

High level commands

Request symboltable command (uses send command byte)

To fetch the entire symboltable from the ECU you will need to issue a command 'S' <CR>
So, two frames are send actually:

Byte	0	1	2	3	4	5	6	7
Data	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x53	0xC4
Data	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x0D	0xC4

Request software version command

To fetch the entire symboltable from the ECU you will need to issue a command 's' <CR>
So, two frames are send actually:

Byte	0	1	2	3	4	5	6	7
Data	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x73	0xC4
Data	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0x0D	0xC4

Write to SRAM command

WAAAADD<CR>
W = 'W' (0x57)
AAAA = address to write to in hexadecimal form in capitals
DD = data to write in hexadecimal for in capitals
<CR> = carriage return character (0x0D)
To issue a write command all bytes in the command stated above should be transmitted as frames.

Example
To write a value 0x1c to SRAM address 0x107a the command to issue would be W107A1C<CR>
The frames to send would then be:

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x57	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x31	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x30	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x37	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x41	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x31	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x43	0xC4
Data	0x00	0x00	0x00	0x00	0x00	0x00	0x0D	0xC4

Read from SRAM command

Byte	0	1	2	3	4	5	6	7
Data	0x00	0x00	0x00	AH	AL	0x00	0x00	0xC7

AH = Address High byte and AL = Address Low byte

The response to this message should be a CAN message with ID 0x0C

Byte	0	1	2	3	4	5	6	7
Data	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	0x0C	0xC6

Monitoring real time data

To be able to monitor your engine's status while driving you can fill the dashboard with gauges (which of course looks very cool) showing you boost pressure, intake air temperature, coolant temperature etc. Another route is to hook up the CAN bus SFI connector to a laptop using the Lawicel CANUSB converter and monitor all engine parameters on the computer. This has several trivial advantages.

- You don't need a gauge for every parameter you want to monitor
- You can monitor much more parameters
- You can rearrange the monitor layout much easier
- You can add logging function for evaluation

T5Suite incorporates a Canbus driver library (for which we owe General Failure lots of gratitude) which enables the application not only to monitor live data from the Trionic, but also to download adaption data from it. This chapter will address the first while the next chapter will address the latter. There's literally hundreds of symbols that can be monitored in Trionic. We will discuss only the most trivial symbols/parameters here. You must be aware of the fact that some symbols consist of multiple 'flags'. These flags indicate a certain state or event in the ECU. Flags are bit masked in a byte or multiple bytes.

Symbol	Description	Type
P_Manifold	Current manifold pressure (kPa)	Byte
Batt_volt	Battery voltage (*10)	Byte
Adapt_injfaktor	Long term fuel adaption factor	Byte
Rpm	Current engine speed	Integer
Rpm_pol	Current engine speed scaled for 1 byte	Byte
Lufttemp	Current manifold intake air temperature	Byte
Kyl_temp	Current cooling water temperature	Byte
Medellast	Average calculated engine load	Byte
Medeltryck	Average inlet manifold pressure	Byte
Medeltrot	Average TPS value	Byte
Lambdaint	Current lambda intergator value	Byte
PWM_ut	PWM signal (%) to BPC valve	Integer
PWM_ut10	PWM signal (%) to BPC valve scaled for 1 byte	Byte
Mis200_tot	Misfire catalyst overheating counter	Integer
Mis1000_tot	Misfire emission degradation counter	Integer
Pgm_status	Program status flags	4/5/6 bytes
Knock_status	Program status flags	4/5/6 bytes
Ready	Flags register indicating ready codes for one trip	Integer
APC_status	APC status byte	Integer
APC_adapt	APC adaption value	Integer
Diag_status	Diagnose status byte	12 bytes
*_error	All error counters	Byte
Last	Current engine load	Byte
Last_delta	Load change	Byte
Bil_hast	Current vehicle speed	Byte
Batt_korr	Correction factor for injection depending on voltage	Integer
AD_sond	A/D value for front lambda sonde	Byte
AD_bat	A/D value for battery voltage	Byte
AD_EGR	A/D value for EGR temperature	Byte
AD_knock	A/D value for knocking detection	Byte
AD_trot	A/D value for Throttle Position Sensor	Byte
AD_lufttemp	A/D value for manifold air intake temperature	Byte
AD_kylv	A/D value for coolant temperature	Byte
U_lambda_eng	Front lambda value	Integer
P_fak	APC controller P factor	Integer
I_fak	APC controller I factor	Integer
D_fak	APC controller D factor	Integer
Regl_tryck	Control pressure (target)	Byte
Gear	Current gear	Byte

Symbol	Byte	Mask	Description
APC_status	0	0x80	Car_speed 45-55 km/h
	0	0x40	Car_speed const. between 45-55 km/h
	0	0x20	I-part for APC regulator activated
	0	0x10	Overboost enable
	0	0x08	Brake
	0	0x04	Cruise
	0	0x02	New reduce of pressure allowed
	0	0x01	Allowed to adapt "I_fak_max"
	1	0x80	Torque reduce automatic
	1	0x40	APC adaption disable
Knock_status	0	0x80	
	0	0x40	
	0	0x20	
	0	0x10	Knock indicates to lower boost ?
	0	0x08	Knock indicates to lower boost ?
	0	0x04	
	0	0x02	
	0	0x01	Knock_average > Knock_average_limit (knock detected)
Diag_status	0	0x01	Idle error
	0	0x02	IAT error
	0	0x04	Speed error
	0	0x08	Front lambda sonde error
	0	0x10	Throttle error
	0	0x20	Pressure error
	0	0x80	Water temperature error
	1	0x01	RPM out error
	1	0x10	Front lambda sonde omslag error
	1	0x80	Rear lambda sonde lean ignition
	2	0x02	Cat error (rear lambda)
	2	0x04	Rear lambda sonde lean
	2	0x80	Cooling water temperature activity test ready
	5	0x01	Start_tryck out of range (mapsensor error?)
	5	0x08	Knock levels within boundaries
	5	0x10	Use diagnose for front lambda parameters
	6	0x01	Something wrong with front lambda sonde
	6	0x04	Indicates that front lambda sensor transition test has failed
	8	0x40	Front lambda sonde transition test ready
	8	0x02	Indicates that OBD2 diagnose part is disabled due to high altitude or low temperature
	8	0x10	Rear lambda sonde activity test ready
	8	0x20	Rear lambda sonde activity test approved
	8	0x40	Front lambda sonde activity test ready
	9	0x20	Rear lambda sonde heater test ready
	9	0x40	Front lambda sonde heater test ready
	11	0x10	Rear lambda sonde “peak over 400 mV” test ok
	11	0x40	Indicates that OBD2 diagnose part is disabled due to high altitude or low temperature
	13	0x02	Tank pressure test enabled
	13	0x04	Tank pressure test ok
	13	0x80	Tank pressure air inlet closed
	14	0x04	Indicates tank pressure error
	14	0x20	Indicates large leakage in tank system
	14	0x10	Indicates small leakage in tank system
	14	0x40	Indicates tank pressure sensor shorted to +5V
	14	0x80	Indicates tank pressure sensor shorted to ground
Ready	0	0x8000	Trip ready

	0	0x4000	Warmup cycle ready
	0	0x2000	Driving cycle ready
	0	0x0001	System performed front lambda sonde transition test
Knock_offset	0	0x7FFF	Knock flag
Readiness	0	0x8000	Indicates catalyst monitor readiness state
	0	0x4000	Indicates EVAP purge monitor readiness state
	0	0x2000	Indicates both oxygen sensor monitor readiness state
	0	0x1000	Indicates both oxygen sensor heater monitor readiness state
MError	0	0x80	Indicates momentary error on TPS signal
	6	0x80	Indicates momentary error on vehicle speed signal
	12	0x80	Indicates momentary error on water temperature signal
	18	0x80	Indicates momentary error on MAP signal
	24	0x80	Indicates momentary error on front lambda sensor
	30	0x80	Indicates momentary error on rear lambda sensor
	36	0x80	Indicates momentary error on battery voltage
	42	0x80	Indicates momentary error on intake air temperature
	48	0x80	Indicates momentary error on knock signal
Cat_status	0	0x8000	Set when delta load are within limits
	0	0x4000	Set when cooling water temperature is above limit
	0	0x2000	Set when vehicle speed is within limits
	0	0x1000	Set when fuel system enters closed loop
	0	0x0800	Set when airflow is within limits
	0	0x0400	Set when warm up data is ok
	0	0x0200	Set when system disapproves current oxygen sensor 1
	0	0x0020	Set when a stage 1 evaluation should start
	0	0x0010	Set if catalyst test fails
	0	0x0008	Set when catalyst evaluation is ready
	0	0x0004	Used to enable catalyst monitor function
	0	0x0001	Set if one or more of the used input sensors has detected errors. Disables catalyst monitor.
	2	0x8000	Set if throttle is open
PMProt	0	0x8000	Purge monitor diagnostic is ready, resets on ignition OFF
	0	0x4000	Indicates that purge monitor test is enabled and currently running
	0	0x2000	Indicates purge monitor test result. Set at ignition OFF. Resets after one successful test.
	0	0x1000	Indicates "Save Lambda Reference" state.
	0	0x0400	Indicates "Ramp To Start" state.
	0	0x0200	Indicates "Perform Test Open" state.
	0	0x0100	Indicates "Perform Test Close" state.
	0	0x0040	Indicates that the system has detected idle state.
	0	0x0020	Indicates that vehicle speed equals 0.
	0	0x0010	Indicated that the system has enabled purge control.
	0	0x0010	Indicated that the system has enabled purge control.
	0	0x0008	Indicates that no failure has been detected on used input sensors.
	0	0x0004	Indicates that purge monitor has entered halt state.
	0	0x0002	Indicates that front lambda transition check is ready.
Diag_mod	0	0x20	Tank pressure diagnostics
Pgm_status	0	0x01	Ignition key on
	0	0x02	Afterstart 2 completed
	0	0x04	Engine is stopped
	0	0x08	Engine is started
	0	0x10	Engine is warm
	0	0x20	Pressure guard triggered (fuel cut)
	0	0x40	Temperature compensation active

	0	0x80	RPM guard is triggered (RPM limiter)
	1	0x01	Application sync ok
	1	0x02	Knock fuel map activated (also see: Knock_status)
	1	0x04	Throttle closed
	1	0x08	Room temperature start
	1	0x10	Fuel off cylinder 4
	1	0x20	Fuel off cylinder 3
	1	0x40	Fuel off cylinder 2
	1	0x80	Fuel off cylinder 1
	2	0x01	Fuel not off (during sync. off ign.)
	2	0x02	Dec.enleanment completed throttledec
	2	0x04	Acc.enrichment completed throttleinc.
	2	0x08	Decrease of retard enrichment allowed
	2	0x10	Start of retard enrichment in progress
	2	0x20	Adaption allowed
	2	0x40	Limp-home mode
	2	0x80	Always active temp.compensation
	3	0x01	Restart
	3	0x02	Active lambda control
	3	0x04	Afterstart enrichment completed
	3	0x08	Init during start completed
	3	0x10	Cooling water enrichment finished
	3	0x20	Purge control active (engine running in closed loop)
	3	0x40	Separate fuel map during idle
	3	0x80	Ignition synchronized
	4	0x01	Sond heating, second sond
	4	0x02	Sond heating, first sond
	4	0x04	ETS error
	4	0x08	Ordinary idle control disable
	4	0x10	Fuel cut allowed (Dashpot)
	4	0x20	Enrichment after fuelcut
	4	0x40	Fulload enrichment
	4	0x80	Fuel syncronized
Idle_status	0	0x80	Neutral position
	0	0x40	AC off
	0	0x20	Engine is running in idle mode
Knock_status	0	0x01	Knock indicator
		0x08	Knock adapted ???
Airpump_status	0	0x80	Airpump output pin 52 on
Restart_status	0	0x08	Soak five minutes

Layout for Error_frame symbol

Offset	Content	Description	Example	Length
0x00	Battery_time	Last battery ok time	0x00 0x00 0x03 0xA7	4
0x04	Diag_tid	Ignition on timer in seconds	0x00 0x00 0x03 0xA8	4
0x08	Hast_tid_grans		0x00 0x00 0x03 0xA7	4
0x0C	I_front_time		0x00 0x00 0x03 0xA8	4
0x10	I_rear_time		0x00 0x00 0x00 0x00	4
0x14	Ign_counter	Ignition counter	0x00 0x00 0xEA 0x55	4
0x18	Knock_time_low		0x00 0x00 0x03 0xA7	4
0x1C	Kyl_tid_end	Last time coolant temp test ok.	0x00 0x00 0x03 0xA8	4
0x20	Luft_tid_end	Last time iat test ok	0x00 0x00 0x03 0xA8	4
0x24	Luft_tid_konst	Time when constant reading of iat was ok	0x00 0x00 0x03 0x7D	4
0x28	Purge_time	Time when purge reading was last ok	0x00 0x00 0x03 0xA7	4
0x2C	Rsond_tid_end	Time when read sonde reading was last ok	0x00 0x00 0x00 0x00	4

0x30	Sond_tid_end		0x00 0x00 0x03 0xA8	4
0x34	Sond_tid_konst		0x00 0x00 0x03 0x93	4
0x38	Tid_kylv_ok		0x00 0x00 0x01 0x91	4
0x3C	Trott_tid_end		0x00 0x00 0x03 0xA8	4
0x40	Tryck_tid_end		0x00 0x00 0x03 0xA8	4
0x44	Tryck_tid_konst		0x00 0x00 0x03 0xA7	4
0x48	unk_34A2		0xFF 0xEF	2
0x4A	unk_3478		0x00 0x1F	2
0x4C	unk_347A		0x03 0x31	2
0x4E	unk_3338		0x00 0x00	2
0x50	unk_333A		0x02 0xC3	2
0x52	Knock_diag_level	Knock diagnostic level	0x01 0x14	2
0x54	Lls_lage		0x01 0x34	2
0x56	Mis200_tot	Misfire catalyst overheat counter (total)	0x00 0x00	2
0x58	Mis200_1	#cyl1	0x00 0x00	2
0x5A	Mis200_2	#cyl2	0x00 0x00	2
0x5C	Mis200_3	#cyl3	0x00 0x00	2
0x5E	Mis200_4	#cyl4	0x00 0x00	2
0x60	Mis1000_tot	Misfire emission level counter (total)	0x00 0x00	2
0x62	Mis1000_1	#cyl1	0x00 0x00	2
0x64	Mis1000_2	#cyl2	0x00 0x00	2
0x66	Mis1000_3	#cyl3	0x00 0x00	2
0x68	Mis1000_4	#cyl4	0x00 0x00	2
0x6A	R_sond_lean_ign		0x00 0x01	2
0x6C	R_sond_rich_ign		0x00 0x00	2
0x6E	Rpm	Engine speed	0x03 0x5C	2
0x70	Run_hours	System uptime hours	0x00 0x04	2
0x72	Sond_omsl_counter		0x00 0x10	2
0x74	Ad_sond	Front lambda AD value	0x28	1
0x75	Ad_cat	Rear lambda AD value	0x01	1
0x76	unk_3474		0x13	1
0x77	unk_3472		0x50	1
0x78	unk_347C		0x22	1
0x79	unk_3476		0x23	1
0x7A	unk_348E		0xFF	1
0x7B	Adapt_injfaktor	Long term fuel trim value	0x80	1
0x7C	Batt_volt	Battery voltage	0x85	1
0x7D	Bil_hast	Vehicle speed	0x00	1
0x7E	Diag_status		0x08 0x00 0x84 0xC0 0x02 0x01 0x00 0x00 0x00	Length
0x86	unk_349E		0x00	1
0x87	Lambdaint	Lambda integrator value	0x23	1
0x88	Medellast	Average load	0x00	1
0x89	unk_34A0		0x22	1
0x8A	P_Manifold	MAP pressure	0x4F	1
0x8B	Purge_value		0x1B	1
0x8C	Pgm_status		0x0C 0x01 0xDE 0x93 0x00 0x02	Length
0x92	Run_minutes	System uptime minutes	0x34	1
0x93	Run_seconds	System uptime seconds	0xFF	1
0x92	Obd_fuel_status	Bug! Same address as Run_minutes		1
0x94	unk_348A		0x7B	1
0x95	unk_348C		0x58	1
0x96	Start_tryck	Starting pressure (ambient?)	0x21	1
0x97	Trot_min		0x61	1
0x98	Kyl_temp	Coolant temperature	0x28	1
0x99	Lufttemp	Intake air temperature	0x15	1
0x9A	unk_3490		0x00	1
0x9B	Airpump_value			1

Downloading adaption data

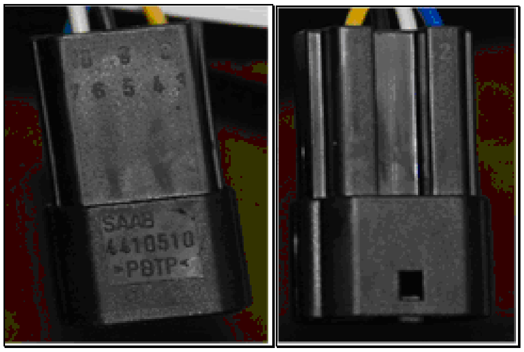
Besides realtime information about the processes inside the engine, the CAN bus interface can also deliver adaption data. This adaption data (tables and variables) is downloaded from the Trionic in a batch method. The user chooses to download the adaption data from the ECU at a given point in time and the adaption data is extracted. With this data we can determine whether or not our mapping is correct because we can see if the Trionic has adapted (positively or negatively) maps to get a better A/F ratio for example. In the near future T5Suite can alter – if the user chooses to – the binary file on the computer with this adaption data automatically. If it turns out that the Trionic has adapted up to its limit something must be wrong and T5Suite will alert the user of this fact. Remapping of the table at hand will in this case be necessary.

Symbol	Description
Apc_adapt	APC controller adaption value. Indicates whether or not boost request values are within limits. If the adaption value is larger, more investigation is needed.
Adapt_korr	Fuel injection adaption table. This table can be downloaded to investigate validity of the initial injection table (Insp_mat). If the values in Adapt_korr reach the limits (Adapt_korr_low or Adapt_korr_high) the initial injection matrix should definitely be remapped or there is something wrong hardware-wise, like faulty fuel pump or regulator.
Adapt_injfaktor	Global injection adaption factor. If this values reaches the limits (Adapt_injfaktor_low or Adapt_injfaktor_high) the fuel injection matrix should definitely be remapped or there is something wrong hardware-wise, like faulty fuel pump or regulator.
Adapt_ind_mat	This map indicates where adaptions have been made to the injection adaption matrix.
Ign_offset_adapt	General ignition adaption offset.
Adapt_ggr	This map contains the number of adaptions made to a certain rpm/load point. If the values in this table are high, Trionic needed to adapt the matrix frequently and something might be wrong.
Adapt_error	Indicates the number of errors detected while adapting.
Adapt_inj_imat	Idle injection map adaption.

Cable example

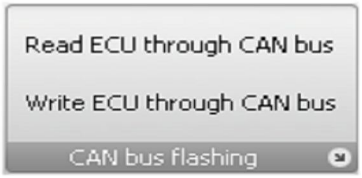
The picture below will give you an example of what a homemade interface cable for a **9000** may look like. Notice that the cable is protected by a plastic wrapper. Two of the four cables inside the wrapper are twisted as mentioned before in the chapter, CANH and CANL. The other two are for ground and programming voltage (+15 V).

The two picture below show the embossed numbers on the connector housing. Numbers 1 and 2 are on one side while 3 to 10 are on the other side.

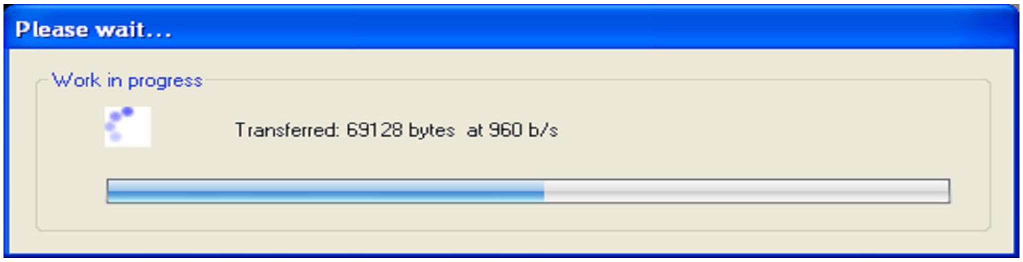


Downloading and flashing Trionic 5

Trionic 5 supports reading and writing to the flash chips in the ECU. To be able to download or upload a program to the ECU a so-called bootloader needs to be uploaded to the ECUs SRAM and executed. Once the bootloader is executed, the program in flash is no longer needed to keep the ECU running and can be erased and reprogrammed. T5Suite has these functions built-in and they are accessible through the menu "Programming functions" in "Read ECU through CAN bus" and "Write ECU through CAN bus".



Once an option has been chosen (assuming all hardware is connected properly) the canbus connection is opened and reading or writing commences.


Please wait...

Work in progress

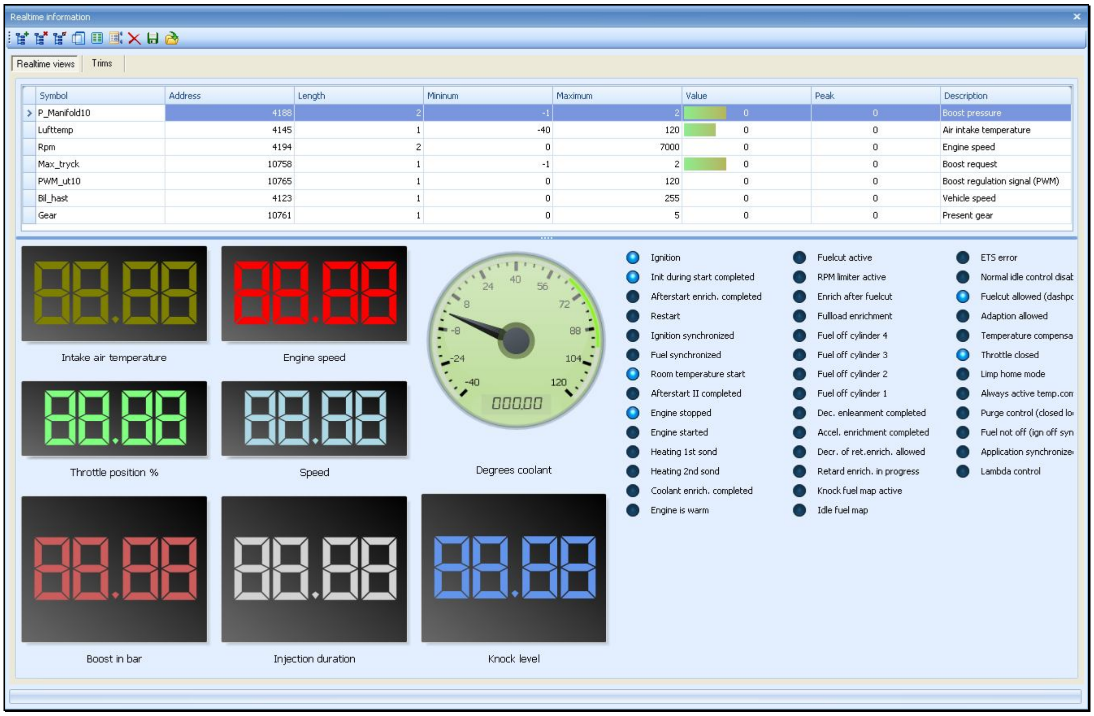
Transferred: 69128 bytes at 960 b/s

Implementation in T5Suite

T5Suite can connect to the ECU through the Lawicel CANUSB adapter using the components delivered with the adapter. The interface implements these functions:

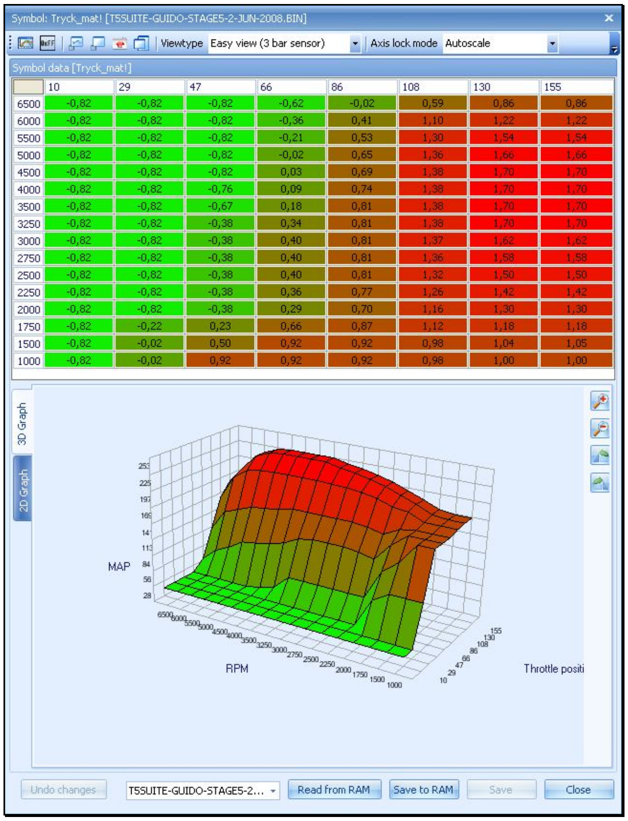
- Retrieving the entire symboltable including SRAM addresses
- Retrieving and saving the adaption data (see paragraph Downloading Adaption data)
- Retrieving specific information at a high update rate (real time viewing and logging)
- Editing variables and tables at runtime in Trionic's SRAM

All realtime data is displayed in a separate panel. This panel can be customized in various ways. The user can select the symbols (variables) to monitor at runtime and the panels entire user interface can be customized. This mean gauges and views can be added and removed from the panel to the users likings. The more symbols are monitored, the more time and processor consuming the process will be. The image below shows the realtime panel in action.



Symbol	Address	Length	Mininum	Maximum	Value	Peak	Description
P_Manifold10	4188	2	-1	2	0	0	Boost pressure
Lufttemp	4145	1	-40	120	0	0	Air intake temperature
Rpm	4194	2	0	7000	0	0	Engine speed
Max_tryck	10758	1	-1	2	0	0	Boost request
PWM_ut10	10765	1	0	120	0	0	Boost regulation signal (PWM)
Bil_hast	4123	1	0	255	0	0	Vehicle speed
Gear	10761	1	0	5	0	0	Present gear

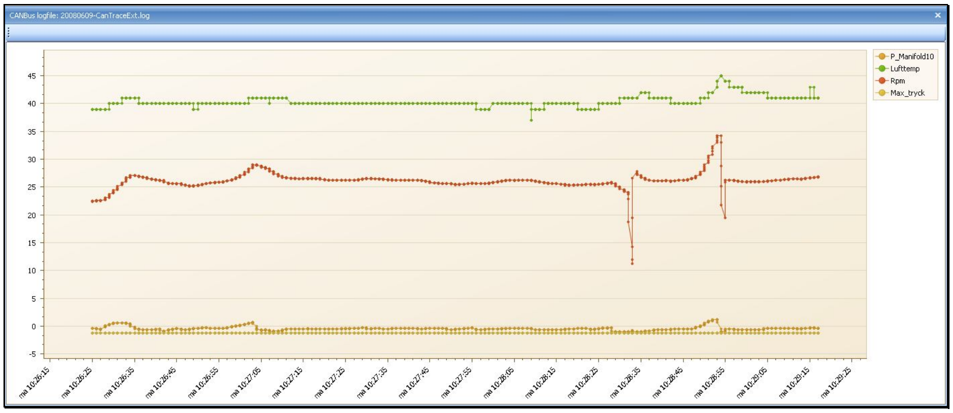
T5Suite also allows for realtime editing of maps in SRAM. This enables to user to tweak the maps without having to reflash between every modification of the software. If the CAN bus interface is connected (realtime panel is opened and CAN connection is running) the user can double click on a map in the symbollist and the corresponding mapviewer will appear.



	10	29	47	66	86	108	130	155
6500	-0,82	-0,82	-0,82	-0,62	-0,02	0,59	0,86	0,86
6000	-0,82	-0,82	-0,82	-0,36	0,41	1,10	1,22	1,22
5500	-0,82	-0,82	-0,82	-0,21	0,53	1,30	1,54	1,54
5000	-0,82	-0,82	-0,82	-0,02	0,65	1,36	1,66	1,66
4500	-0,82	-0,82	-0,82	0,03	0,69	1,38	1,70	1,70
4000	-0,82	-0,82	-0,76	0,09	0,74	1,38	1,70	1,70
3500	-0,82	-0,82	-0,67	0,18	0,81	1,38	1,70	1,70
3250	-0,82	-0,82	-0,38	0,34	0,81	1,38	1,70	1,70
3000	-0,82	-0,82	-0,38	0,40	0,81	1,37	1,62	1,62
2750	-0,82	-0,82	-0,38	0,40	0,81	1,36	1,58	1,58
2500	-0,82	-0,82	-0,38	0,40	0,81	1,32	1,50	1,50
2250	-0,82	-0,82	-0,38	0,36	0,77	1,26	1,42	1,42
2000	-0,82	-0,82	-0,38	0,29	0,70	1,16	1,30	1,30
1750	-0,82	-0,22	0,23	0,66	0,87	1,12	1,18	1,18
1500	-0,82	-0,02	0,50	0,92	0,92	0,98	1,04	1,05
1000	-0,82	-0,02	0,92	0,92	0,92	0,98	1,00	1,00

Now we see a set of extra buttons that allows the user to get the maps data from SRAM and write it back again. Please note that altering maps while driving the car is very, VERY dangerous. Please only edit maps when the car is parked at a safe spot!!

All data the is requested by the CAN interface (items entered in the realtime view) will also be logged to file. It is possible to view this data in T5Suite as well as export it to LogWorks (http://www.innovatemotorsports.com/support.php). Within T5Suite the log can look something like the image below.



Common mistakes and FAQ

General

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

Fuel knock map

When knocking occurs inside the engine (sensed through the DI cartridge) evenly in all cylinders a tiny bit of extra fuel will be injected. When the manifold pressure (MAP) is over 140 kPa the normal injection map will be overruled by another map: the knocking fuel map. This map enriches the fuel mixture so that the engine cools down (extra fuel means cooler mixture).

A common mistake in changing maps is that the injection map is altered but the knocking map is forgotten. This may result in lower (or equal) values in the knocking map than in the normal injection map. When the engine starts to knock and the map is switched to supply more fuel to cool the engine, not more but less fuel will be injected. This, of course, is a nightmare for your engine.

Ignition advance

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

Ignition retard

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

Injector constant

After changing the stock injectors for larger ones, the injector constant has to be changed. Keep in mind the the injector constant needs to be over ~10 to keep a good resolution for fuel injection time calculation (630 cc injectors have an injector constant of 14-17 so there no need to go below 10.

Boost request map

Some tuners change the boost request map(s) to absurd values. Sometimes because they think that it's just a request map, so the engine will make maximum power anyway, sometimes because they try to compensate for weak wastegate springs and such.
The boost request map should not be higher than what the engine can achieve. So, putting 1.45 bar request @ 5500 RPM with a Garrett T25 turbine is useless because the little turbo cannot sustain this pressure at higher engine speeds. The only result this will have is socalled negative adaption. The Trionic will notice that the requested boost is not met by the engine and adapt the boost request accordingly. This can even result in lower boost in other load and rpm sites and thus give you the result of less power than can be made at that load / rpm site.

FAQ

Question: At what AFR should I try to keep the engine?
Answer: Try to keep the AFR between 10.8 and 12 @ wide open throttle

Question: At what EGT should I try to keep the engine?
Answer: Try to keep the exhaust gas temperature below 950°C.

Question: What is the approximate ignition advance at wide open throttle?

Answer: Try to keep the ignition advance at approximately 10° BTDC @ wide open throttle and rpm limit

Question: How can I determine what the maximum boost request should be for my turbo?
Answer: 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.

Question: What is a good intake air temperature?
Answer: 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.

Tools

T5Suite

T5Suite was initially build to contain all separate tools available for T5. It grew a little and at the moment of writing this document it has these functions:

- Checksum verification and correction
- Software ID adjustment
- Immobilizer code adjustment
- File comment adjustment
- Box number adjustment
- VSS enable and disable
- Splitting binaries
- Merging binaries
- Extraction of symbol table
- Generation of XDF file for TunerPro
- Map visualization
- Compare maps in binary to another binary.
- Tuning wizard to stage 1, 2, 3 and X (X means free parameter input)
- Modify binary for use with 3 bar mapsensor
- Modify injector constant for use with larger injectors
- Check binary for anomalies (see next chapter)
- Realtime monitoring of data
- Realtime mapping of data
- Logging in T5Suite format and exporting to LogWorks©

For usage of this tool please refer to its user manual.

BD32

BD32.exe is the tool used to interface with the ECU through a BDM interface. It is DOS based and will run normally on Win95/98/Me. Ideally the user would boot into a DOS environment to use the tool. There is a windows version available but the author has no experience with that specific tool.

IDA Pro

IDAPro stands for Interactive DisAssembler Professional. It enables the user to disassemble binary files to its original source code. IDAPro is commercial software, not freeware.

Example of how to use IDA Pro with a **Trionic 7** box.
Open binary/raw file
Set processor to Motorola series: 68330
Check the 'Create RAM section', start address 0xF00000, size 0xFFFF.
Go to address ROM:00000000 in 'IDA View-A' and hit D-key three times to get dc.l $FFFFEFFC
Go to the next address ROM:00000004 and hit D-key three times to get reset vector address (this varies from binary to binary)
You should get e.g. dc.l unk_5169A or something like that, double-click the unk_5169A text
Your now in the place where the code execution starts, press C to disassemble
Now, from the menu select Options -> General, go to Analysis tab and press 'Kernel options1' button

Check 'Make final analysis pass' and hit OK
Press 'Reanalyze program' button and wait a while (this really takes some time, a minute or so)

Hex editor

Hexworkshop (or UltraEdit) is a tool that comes in handy often. It can be used to view, search and modify the raw binary file.

References

Webreferences

- ECUproject initiative [www.ecuprojct.com Steve Hayes and friends]
- Xendus [www.xendus.se General Failure]
- SaabCentral [www.saabcentral.com]
- Motorola datasheet on MC68332
- Trionic Wiki pages [http://en.wikipedia.org/wiki/Trionic]
- T5Suite homepage [http://trionic.mobixs.eu]
- Townsendimports [www.townsendimports.com]
- BDM Software [http://trionic.mobixs.eu/bd32-122.zip]
- T5 scripts including T5.2 [http://trionic.mobixs.eu/FLASHT5ANDT52.zip]
- Ion sensing for knock detection [http://www.fs.isy.liu.se/~larer/Projects/main.html]
- Turbo compressor maps [http://www.automotivearticles.com/Turbo_Selection.shtml]
- Saab9000.com [http://www.saab9000.com]
- JKBPower forum [http://jkbpower.egetforum.se/forum/index.php]

Appendix I : Symbol list

Flash symbols

Symbol name	Category	Subcategory	Description
AC_const!	Misc	Advanced	Number of half enginerevs AC compensation is active.
AC_Control!	Misc	Advanced	Data to control the ac-compressor when the brake pedal is pressed. This function must be enabled with a bit in Pgm_mod.
AC_fak!	Sensor	Lambda sensor	Additive (enrichment) offset to lambdacontrol when AC activates.
AC_slope!	Misc	Advanced	Number of half enginerevs between every decreament of AC_fak when ramping if lamdacontrol is not activated.
AC_start_tid!	Idle	Advanced	AC starttime
AC_wait_off!	Idle	Advanced	Delaytime in milliseconds when AC disables. Function of air temperature
AC_wait_on!	Idle	Advanced	Delaytime in milliseconds when AC engages. Function of air temperature
Accel_konst!	Fuel	Transient	Fuel Injection Map, Acceleration, Throttle Position.
Accel_temp!	Fuel	Temperature compensation	Acceleration temperature compensation
Accel_temp2!	Fuel	Temperature compensation	Acceleration temperature compensation #2
Adapt_inj_imat!	Fuel	Idle	Idle Injection map adaption.
Adapt_injfaktor!	Fuel	Advanced	Long term fuel adaption factor.
Adapt_injfaktor_high!	Fuel	Advanced	If the long term fuel adaption (Adapt_injfaktor) is above Adapt_injfaktor_high, adaption error is set. Resolution is 0.008.
Adapt_injfaktor_low!	Fuel	Advanced	If the long term fuel adaption (Adapt_injfaktor) is below Adapt_injfaktor_low, adaption error is set. Resolution is 0.008.
Adapt_korr!	Fuel	Advanced	The adapted injection table (spot adaption base)
Adapt_korr_high!	Fuel	Advanced	Maximum value for fuel adaption map
Adapt_korr_low!	Fuel	Advanced	Minimal value for fuel adaption map
AdjustLamCal!	Sensor	Lambda sensor	Lean and rich step adjustment depending on rear lambda sensor
After_fcut_tab!	Fuel	Enrichment	After fuelcut enrichment (function of number of combustions after fuelcut)
Airpump_ign_offset!	Ignition	Advanced	Ignitionangle offset value when the airpump are activated. A negative value will decrease the ignitionangle.
Airpump_inj_factor!	Fuel	Advanced	Injectiontime factor when the airpump are activated. A value above 1.00 will increase the injectiontime.
Ap_lambda_delay!	Sensor	Lambda sensor	Delaytime between that the airpump are turned off and that the lamdacontrol are activated.
Ap_max_on_time!	Misc	Airpump control	Maximum time the airpump will bee activated after the engine are started.
Ap_max_on_watertemp!	Misc	Airpump control	Maximum watertemperature for the airpump. If the watertemperature are above this limit will the airpump bee turned off.
Ap_max_rpm!	Misc	Airpump control	Maximum engine speed for the airpump. If the rpm are above this limit will the airpump bee turned off.
Ap_max_start_airtemp!	Misc	Airpump control	Maximum start airtemperature for the airpump. If the start airtemperature are above this limit will the airpump never turn on.
Ap_max_start_watertemp!	Misc	Airpump control	Maximum start watertemperature for the airpump. If the start watertemperature are above this limit will the airpump never turn on.
Ap_min_airtemp!	Misc	Airpump control	Minimum start airtemperature for the airpump. If the start airtemperature are below this limit will the airpump never turn on.
Apc_cert_tid!	Boost control	Advanced	Apc_cert_tid
Apc_decrese_asp!	Boost control	Advanced	Apc_decrese_asp
Apc_frek!	Boost control	Advanced	Apc control frequency
Apc_fullgas!	Boost control	Advanced	Apc_fullgas
Apc_knock_tab!	Boost control	Advanced	Apc_knock_tab
Apc_konst_tid!	Boost control	Advanced	Apc_konst_tid

API_dt_delay!	Misc	Advanced automatic	Delay time for 'drive and throttle' ignition retardation before ramping back.
API_dt_offset_tab!	Ignition	Advanced automatic	Function to reduce torque when driver switches from NEUTRAL to DRIVE and has activated accelerator pedal.
API_dt_ramp!	Misc	Advanced automatic	Ramp time for 'drive and throttle' ignition retardation ramping back.
API_ku_delay!	Ignition	Advanced automatic	Delay time for 'kickdown and upshift' ignition retardation before ramping back.
API_ku_derivata!	Ignition	Advanced automatic	Negative rpm derivata limit (rpm/10ms).
API_ku_offset!	Ignition	Advanced automatic	Function to reduce torque when driver kick's down and automatic gearbox shifts up.
API_ku_ramp!	Ignition	Advanced automatic	Ramp time for 'kickdown and upshift' ignition retardation ramping back.
API_rpm_limit!	Misc	Advanced automatic	Below this rpm limit no 'kickdown and upshift' ignition retardation will be enabled.
API_throttle_limit!	Ignition	Advanced automatic	Below this throttle limit no 'kickdown and upshift' ignition retardation will be enabled.
API_throttle_tab!	Misc	Advanced automatic	Table used to determine ignition offset retardation depending on relative throttle angle when DRIVE is engaged.
Batt_korr_tab!	Fuel	Advanced	Correction on injectiontime depending on battery voltage
Before_start!	Fuel	Cranking	Parallel injection on all cylinders before engine start to aid in starting a cold engine.
Br_minus_tab!	Fuel	Idle	Decrease of fuel during idle when rpm < rpm nominal. Resolution is uS
Br_plus_tab!	Fuel	Idle	Increase of fuel during idle when rpm < rpm nominal. Resolution is uS
BS_temp!	Misc	Advanced	Enables before start at restart after 0 to 45 sec if last start was succsessful and coolingwater temperature is above BS_temp.
Build_up_time!	Fuel	Cranking	Delay between first detected crank tooth and before start. Compensation for building up fuel system pressure. Active when coolingwater temp is above Build_up_time_switch.
Build_up_time_2!	Fuel	Cranking	Delay between first detected crank tooth and before start. Compensation for building up fuel system pressure. Active when coolingwater temp is below Build_up_time_switch.
Build_up_time_switch!	Fuel	Cranking	Temp limit for Build_up_time. if(Kyl_temp > Build_up_time_switch) Use Build_up_time for delay. else Use Build_up_time_2 for delay.
Cat_af_start_timer_min!	Sensor	Lambda sensor	If the air flow is below this threshold, the Cat_start_timer halts.
Cat_air_flow_hi!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Cat_air_flow > Cat_air_flow_hi) Catalyst monitor disables. Resolution is 1.
Cat_air_flow_lo!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Cat_air_flow < Cat_air_flow_lo) Catalyst monitor disables. Resolution is 1.
Cat_air_flow_tab!	Sensor	Lambda sensor	Airflow site for calibrating tables using airflow as input data. Resolution is 1.
Cat_cool_temp_lim!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Kyl_temp < Cat_cool_temp_lim) Catalyst monitor disables. Resolution is 1øC.
Cat_delta_load_lim!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Cat_delta_load > Cat_delta_load_lim) Catalyst monitor disables. Resolution is 1 kPa.
Cat_load_filt_coef!	Sensor	Lambda sensor	To average the engine load value.(Last). Used to detect load transients. 0.000 => Max filtering. 1.000 => Min filtering. filter_coeficient = 1-exp(-loop_time / time_constant) loop_time = 0.010s Resolution is 0.001.
Cat_load_tab!	Sensor	Lambda sensor	Load site for calibrating maps using load as input data. Resolution is 1.
Cat_monitor_restart!	Sensor	Lambda sensor	If set to one, the catalyst monitor will restart. (automaticly set to zero directly afterwards)
Cat_ox1_bias_lim!	Sensor	Lambda sensor	The O_2 bias is the point where the O_2 sensor switches from rich to lean. This value is substracted from the filtrated O_2 voltage to get Cat_ox1_err. Resolution is 1 mV..
Cat_ox1_duty_hi_lim!	Sensor	Lambda sensor	To define the maximum acceptable duty cycle of oxygen sensor 1. if (Cat_ox1_duty_cycle > Cat_ox1_duty_hi_lim) Do not use current cycle data for catalyst monitor. Resolution is 1%.
Cat_ox1_duty_lo_lim!	Sensor	Lambda sensor	To define the minimum acceptable duty cycle of oxygen sensor 1. if (Cat_ox1_duty_cycle < Cat_ox1_duty_lo_lim) Do not use current cycle data for catalyst monitor. Resolution is 1%.

Cat_ox1_err_lim!	Sensor	Lambda sensor	Its purpose is to assure that a stable fuel control condition exists before data can be considered valid for test purpose. if (Cat_ox1_err > Cat_ox1_err_lim) Do not use current cycle data for catalyst monitor. Resolution is 1 mV..
Cat_ox1_err_max_tab!	Sensor	Lambda sensor	Table to control Cat_ox2_dev_max_lim. Input variable Cat_ox1_max_err. Resolution is 1mV..
Cat_ox1_filt_coef_map!	Sensor	Lambda sensor	Look-up tabe of filter coefficients as a function of load and rpm. As driving conditions changes, so does the O_2 frequency, therefore the amount of filtering will need to change to maintain a fixed amplitude filtered output. This table look-up will provide a varying coefficient to keep the amount of filtering constant at all points. Resolution is 0.001.
Cat_ox1_per_hi_tab!	Sensor	Lambda sensor	To eliminate catalyst oxygen sensor data whose high period is higher (longer) than expected. This look-up table is a function of engine airflow. Resolution is 1 mS..
Cat_ox1_per_lo_tab!	Sensor	Lambda sensor	To eliminate catalyst oxygen sensor data whose low period is higher (longer) than expected. This look-up table is a function of engine airflow. Resolution is 1 mS..
Cat_ox2_dev_max_map!	Sensor	Lambda sensor	Table look-up as function of load and rpm. This calibration map defines a threshold for the Cat_ox2_dev_max_lim. Cat_dev_diff = Cat_ox2_dev_max - Cat_ox2_dev_max_lim Resolution is 1.
Cat_ox2_dev_samp_lim!	Sensor	Lambda sensor	To determine if enough samples have been collected to average the post-catalyst oxygen sensor maximum deviation data. if (Cat_ox2_dev_samp >= Cat_ox2_dev_samp_lim) Cat_avg_dev_diff = Cat_ox2_dev_sum / Cat_ox2_dev_samp Resolution is 1.
Cat_ox2_filt_coef_2!	Sensor	Lambda sensor	Filter coeficient to second LP-filter on oxygen sensor 2 signal. (used to create Cat_ox2_dev)
Cat_ox2_filt_coef_map!	Sensor	Lambda sensor	Look-up tabe of filter coefficients as a function of load and rpm. As driving conditions changes, so does the O_2 frequency, therefore the amount of filtering will need to change to maintain a fixed amplitude filtered output. This table look-up will provide a varying coefficient to keep the amount of filtering constant at all points. Resolution is 0.001.
Cat_ready_timer_lim!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Cat_ready_timer = Cat_ready_timer_lim) Catalyst monitor enables. Resolution is 1 sec..
Cat_ready_timer_step!	Sensor	Lambda sensor	Used to increase recovering period for catalyst after fuel dependent actions. Cat_ready_timer_step is decremented from the ready timer if the warm up data are met and the warm up data breaks due to high delta load or lambda ctrl. deactivation.
Cat_rpm_tab!	Sensor	Lambda sensor	Rpm site for calibrating maps using engine speed as input data. Resolution is 10.
Cat_stage1_threshold!	Sensor	Lambda sensor	To determine if the post-catalyst oxygen sensor's average deviation difference is small enough to pass the diagnostic test. if (Cat_avg_dev_diff > Cat_stage1_threshold) catalyst converter test fails. else catalyst converter test passes. Resolution is 1.
Cat_start_timer_lim!	Sensor	Lambda sensor	To enable catalyst monitor. if (Cat_start_timer < Cat_start_timer_lim) Catalyst monitor disabled.
Cat_vehi_speed_hi!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Bil_hast > Cat_vehi_speed_hi) Catalyst monitor disables. Resolution is 1 km/h.
Cat_vehi_speed_lo!	Sensor	Lambda sensor	Used to enable catalyst monitor function. if (Bil_hast < Cat_vehi_speed_lo) Catalyst monitor disables. Resolution is 1 km/h.
Cut_ej_under!	Fuel	Advanced	Lower engine speed limit to be able to allow fuel cut
Cut_temp!	Fuel	Advanced	Minimal coolingwatertemp to allow fuel cut
Cyl_komp!	Fuel	Advanced	Limit for cylinder detection low part Fuel compensation factor due to differences between cylinders.
D_fors!	Boost control	Advanced manual	Boost Regulation Map, D-Gain factor of PID control
D_fors_a!	Boost control	Advanced automatic	Boost Regulation Map, D-Gain factor of PID control (automatic)
Dash_act_step!	Idle	Advanced	Dash act step
Dash_rpm_axis!	Fuel	Axis	Dash pot tabel rpm axis
Dash_tab!	Fuel	Advanced	Dash pot position interpolated from matrix.
Dash_trot_axis!	Fuel	Axis	Dash pot tabel rpm axis
Data_namn!	Misc	Basic	Software version
Del_mat!	Fuel	Advanced	Delay map (crankshaft degrees)

Delay!	Idle	Advanced	Number of 131 millisecond intervals to permit adaption. (How many seconds engine should be in the I part to allow adaption)
Derivata_br_sp!	Fuel	Axis	Fuel adjustment map support points
Derivata_br_tab_neg!	Fuel	Idle	Fuel adjustment, negative rpm delta
Derivata_br_tab_pos!	Fuel	Idle	Fuel adjustment, positive rpm delta
Derivata_fuel_rpm!	Fuel	Idle	Rpm limit for fuel adjustment
Derivata_grans!	Idle	Advanced	Smallest rpm difference for idle control
Detect_map!	Diagnostics	Misfire	Reference map for misfire detection. Resolution is 1.
Detect_map_x_axis!	Diagnostics	Axis	Control's the current working position in Detect_map depending on inlet manifold pressure. Resolution is 1.
Detect_map_y_axis!	Diagnostics	Axis	Control's the current working position in Detect_map depending on engine speed. Resolution is 10.
DI_cassette_U_limit!	Diagnostics	Advanced	Below this voltage, signals from the DI-cassette can not be trusted. Used to disable some diagnostic tests to prevent corruption depending on bad input signals. Resolution is 0.1 V.
Diag_speed_load!	Diagnostics	Advanced	Threshold for detecting missing vehicle speed signal. (Last > Diag_speed_load) Resolution is 1.
Diag_speed_rpm!	Diagnostics	Advanced	Threshold for detecting missing vehicle speed signal. (RPM > Diag_speed_rpm) Resolution is 10 rpm.
Diag_speed_time!	Misc	Advanced	Timelimit for load and rpm to be over their limits for detecting vehicle speed error.
Eft_dec_fak!	Fuel	Warmup	Number of half enginerev. between a unit of decrease for Eft_fak1. Number of half enginerev is temperature dependent.
Eft_dec_fak2!	Fuel	Warmup	Number of half enginerev. between a unit of decrease for Eft_fak2. Number of half enginerev is temperature dependent.
Eft_fak_minsk!	Fuel	Warmup	Number of steps to decrease Eft_fak2
Eftersta_fak!	Fuel	Warmup	Enrichmentfactor for coldstart #1. Factor is depending on coolantwater temperature.
Eftersta_fak2!	Fuel	Warmup	Enrichmentfactor for coldstart #2. Factor is depending on coolantwater temperature.
Expand!	Misc	Advanced	Multiplicative factor to load and engine-speed axles. Introduced to experiment with idlemap.
Fload_tab!	Fuel	Enrichment	Full load map for full load enrichment
Fload_throt_tab!	Fuel	Enrichment	Full load throttle map for full load enrichment
Frekv_map!	Sensor	Lambda sensor	Lambda ramp combustion delay
Fuel_knock_mat!	Fuel	Enrichment	Fuel enrichment table when knocking occurs
Fuel_knock_xaxis!	Fuel	Axis	Fuel knock correction table x axis
Fuel_map_xaxis!	Fuel	Axis	Fuel correction table x axis
Fuel_map_yaxis!	Fuel	Axis	Fuel correction table y axis
Gear_ratio_delta!	Runtime	Basic	Car speed calculation, gear ratio delta
Gear_st!	Runtime	Basic	Car speed calculation, gear stages
Halva_sidan!	Fuel	Advanced	This constant is used by pointadaption to decide if the riftpoint the engine works in is close enough to the matrixpoint which is closest to allow adaption (0-7 - 7 is easyest to allow adaption)
Hast_grans!	Idle	Advanced	Lowest vehicle speed with idle control. (Minimal car speed limit to enable idle control)
Hot_decr!	Fuel	Enrichment	Decrementtime on Hot_start_fak
Hot_tab!	Fuel	Enrichment	Enrichmentfactor when hotstartenrichment is active
Hp_support_points!	Fuel	Axis	Restart factor for heatplates support points
I_fors!	Boost control	Advanced manual	Boost Regulation Map, I-Gain factor of PID control
I_fors_a!	Boost control	Advanced automatic	Boost Regulation Map, I-Gain factor of PID control (automatic)
I_kyl_st!	Misc	Temperature calculation	Support points for cooling water temperature dependend maps
I_last_rpm!	Idle	Advanced	Value for calculation of nominal load. Function of rpm.
I_last_temp!	Idle	Advanced	Value for calculation of nominal load. Function of coolant water temperature.
I_luft_cel!	Misc	Advanced	Last actual airtemp (more text)
I_luft_st!	Misc	Temperature calculation	Support points for airtemp dependend mas
I_medel_varv!	Idle	Advanced	Number of rotations for integrating control. (Number of engine revolutions needed to calculate average enginespeed)
Idle_ac_extra_sp!	Idle	Axis	Rpm breakpoints for Idle_ac_extra_tab.
Idle_ac_tab!	Idle	Advanced	Drive with AC, function of airtemp. Extra step for idle valve position when AC is turned on.
Idle_der_tab!	Idle	Advanced	(D) Above nominal enginespeed when rpm is decreasing
Idle_drift_tab!	Idle	Advanced	Drive/Neutral value
Idle_ej_off!	Idle	Advanced	Increment of valveposition first time when not idle.

Idle_fuel_korr!	Fuel	Idle	Idle fuel correction map
Idle_p_tab!	Idle	Advanced	(P) Decrement value over nominal rpm
Idle_rpm_off!	Idle	Advanced	Increment of nominal rpm when Drive/Neutral
Idle_rpm_tab!	Idle	Advanced	Nominal rpm depending on cooling water temperature
Idle_st_last!	Fuel	Axis	Injection on idle load axis values.
Idle_st_rpm!	Fuel	Axis	Injection on idle RPM axis values.
Idle_start!	Idle	Advanced	Idle start variable
Idle_start_extra!	Fuel	Idle	Idle start extra, new value after 5 minutes soak
Idle_start_extra_ramp!	Fuel	Idle	Number of 180 degrees pulses for decrement
Idle_start_extra_sp!	Fuel	Axis	Idle start extra, new value after 5 minutes soak, map support points
Idle_step_imns!	Idle	Advanced	Decrement value within the inner band
Idle_step_ioka!	Idle	Advanced	Increment value within the inner band
Idle_temp_off!	Idle	Advanced	Additional factor to valveposition. Function of cooling water temperature)
Idle_tryck!	Fuel	Cranking	Increase of valve position during start. Function of AD value pressure sensor.
Ign_idle_angle!	Ignition	Idle	Target ignition advance for idle state.
Ign_map_0!	Ignition	Basic	Main ignition map for warm engine (0)
Ign_map_0_x_axis!	Ignition	Axis	Ignition map 0 x-axis, pressure from MAP sensor
Ign_map_0_y_axis!	Ignition	Axis	Ignition map 0 y-axis, RPM
Ign_map_1!	Ignition	Idle	Ignition map for idle correction (1)
Ign_map_1_x_axis!	Ignition	Axis	Ignition map 1 x-axis.
Ign_map_1_y_axis!	Ignition	Axis	Ignition map 1 y-axis, RPM deviation
Ign_map_2!	Ignition	Advanced	Ignition map for knocking correction (2)
Ign_map_2_x_axis!	Ignition	Axis	Ignition map 2 x-axis, pressure from MAP sensor
Ign_map_2_y_axis!	Ignition	Axis	Ignition map 2 y-axis, RPM
Ign_map_3!	Ignition	Advanced	Transient ignition map (3)
Ign_map_3_x_axis!	Ignition	Axis	Ignition map 3 x-axis, cooling water AD
Ign_map_3_y_axis!	Ignition	Axis	Ignition map 3 y-axis, RPM
Ign_map_4!	Ignition	Basic	Ignition map for warmup (cool engine) (4)
Ign_map_5!	Ignition	Advanced automatic	Ignition map for stalling engine (5)
Ign_map_5_x_axis!	Ignition	Axis	Ignition map 5 x-axis
Ign_map_5_y_axis!	Ignition	Axis	Ignition map 5 y-axis, RPM
Ign_map_6!	Ignition	Advanced	Ignition map for torque reduction on upshift (6)
Ign_map_6_x_axis!	Ignition	Axis	Ignition map 6 x-axis, pressure from MAP sensor
Ign_map_6_y_axis!	Ignition	Axis	Ignition map 6 y-axis, RPM
Ign_map_7!	Ignition	Advanced	Ignition map for torque reduction on downshift (7)
Ign_map_7_x_axis!	Ignition	Axis	Ignition map 7 x-axis, pressure from MAP sensor
Ign_map_7_y_axis!	Ignition	Axis	Ignition map 7 y-axis, RPM
Ign_map_8!	Ignition	Idle	Ignition map for idle ignition compensation during speed (8)
Ign_map_8_x_axis!	Ignition	Axis	Ignition map 8 x-axis
Ign_map_8_y_axis!	Ignition	Axis	Ignition map 8 y-axis, RPM delta
Ign_offset!	Ignition	Basic	Global ignition offset, added to calculated ignition angle.
Ign_offset_adapt!	Ignition	Advanced	General ignition offset adaption.
Ign_offset_cyl1!	Ignition	Advanced	Individual ignition offset for cylinder 1, added to calculated ignition angle.
Ign_offset_cyl2!	Ignition	Advanced	Individual ignition offset for cylinder 2, added to calculated ignition angle.
Ign_offset_cyl3!	Ignition	Advanced	Individual ignition offset for cylinder 3, added to calculated ignition angle.
Ign_offset_cyl4!	Ignition	Advanced	Individual ignition offset for cylinder 4, added to calculated ignition angle.
Indent_kvot!	Undocumented	Undocumented	Not used.
Inj_konst!	Fuel	Advanced	Injector scaling (adjust for different injectors)
Insp_mat!	Fuel	Basic	Volumetric Efficiencey table (RPM x MAP)
Iv_min_change!	Misc	Advanced	Step for how much adaption should been adjust.
Iv_min_count!	Misc	Advanced	Idle valve counter
Iv_min_load!	Misc	Advanced	Pressure for which adaption should been adjust.
Iv_min_tab!	Fuel	Idle	Idle valve min table.
Iv_min_tab_ac!	Fuel	Idle	Idle valve min table ac.
Iv_start_pos_offset!	Fuel	Idle	Idle valve after start position offset
Iv_start_rpm_offset!	Fuel	Idle	Idle valve after start rpm offset
Iv_start_time_tab!	Fuel	Idle	Time for higher idle speed after start
Knock_lim_tab!	Ignition	Advanced	Maximum allowed knock offset
Knock_press_tab!	Ignition	Advanced	Pressure limit for knock indication
Knock_ref_matrix!	Ignition	Advanced	Knock reference map
Knock_start!	Misc	Advanced	Knock window start time

Knock_wind_off_tab!	Ignition	Advanced	Closing angle of knock window from matrix
Knock_wind_on_tab!	Ignition	Advanced	Opening angle of knock window from matrix
Knock_wind_rpm!	Ignition	Axis	Knock_wind table rpm axis
Konst_lean_map!	Sensor	Lambda sensor	Lean constant lambdaintegrator, combustions
Konst_rich_map!	Sensor	Lambda sensor	Rich constant lambdaintegrator, combustions
Kyltemp_steg!	Misc	Temperature calculation	Map to convert coolingwatertemps A/D value to øC,
Kyltemp_tab!	Misc	Temperature calculation	Same as Kyltemp_steg, Value in øC +40
Lacc_konst!	Fuel	Transient	Fuel Injection Map, Acceleration, Load Delta.
Lam_laststeg!	Sensor	Axis	Lambda control table load steps
Lam_load_sp!	Sensor	Axis	Lambda load support points
Lam_minlast!	Sensor	Lambda sensor	Lambda minimal load
Lam_rpm_sp!	Sensor	Axis	Lambda rpm support points
Lam_rpmsteg!	Sensor	Axis	Lambda control table rpm steps
Lamb_ej!	Sensor	Lambda sensor	Lambda no, temperature limit for open throttle
Lamb_idle!	Sensor	Lambda sensor	Lambda idle, temperature limit for closed throttle
Lamb_kyl!	Sensor	Axis	Lambda map support points for coolant temperature
Lamb_temp!	Sensor	Lambda sensor	Min. Coolingwatertemp to start Lambdacontrol.
Lamb_tid!	Sensor	Lambda sensor	Lambda time (number of combustions)
Lambda_cat_lean!	Sensor	Lambda sensor	Lambda sensor lean value
Lambda_cat_rich!	Sensor	Lambda sensor	Lambda sensor rich value
Lambdamatris_diag!	Sensor	Lambda sensor	Lambda sonde diagnostics control factors
Last_reg_ac!	Idle	Advanced	Value for calculation of nominal load. Function of AC.
Last_reg_kon!	Idle	Advanced	Current reg_kon_mat values
Last_temp_st!	Idle	Axis	Nominal load calculation maps support points
Last_varv_st!	Idle	Axis	Nominal load calculation maps support points
Lean_step_map!	Sensor	Lambda sensor	Lean step for lambda integrator
Leave_shut!	Sensor	Lambda sensor	Additive factor to lambda control when leaving idlecontrol This results in a enrichment when opening throttle
LhomeMapPman!	Misc	Limp home	Limphome map for pressure in manifold (P_manifold)
LhomeXaxisPman!	Misc	Axis	Throttle values for Limphome pressure map
LhomeYaxisPman!	Misc	Axis	RPM values for Limphome pressure map
Limit_high!	Misc	Advanced	Limit for cylinder detection high part
Limit_low!	Misc	Advanced	Limit for cylinder detection low part
Limp_tryck_konst!	Boost control	Advanced	Maximum boost in limp-home mode
Lknock_oref_tab!	Ignition	Advanced	Offset reference table for large knock
Lret_konst!	Fuel	Transient	Fuel Injection Map, Deceleration, Load Delta.
Luft_kompfak!	Misc	Temperature calculation	Map over correction factors on injectiontime as a function of airtemp. Map-support points from Lufttemp_steg
Lufttemp_steg!	Misc	Temperature calculation	Map-support points used in the most airtemp dependend maps
Lufttemp_tab!	Misc	Temperature calculation	Map to convert airtemps A/D value to øC, map-support points in Lufttemp_steg. Value in øC +40.
Max_dead!	Idle	Advanced	Rpm offset for higher integrator limit.
Max_load_gadapt!	Fuel	Advanced	Maximum Load global fuel injection adaption.
Max_ratio_aut!	Boost control	Advanced automatic	Below this limit load pressure reduce is active.
Max_regl_sp!	Boost control	Axis	Maximum boost pressure limit maps support points
Max_regl_temp_1!	Boost control	Advanced	Maximum boost pressure to set above temperature limit #1
Max_regl_temp_2!	Boost control	Advanced	Maximum boost pressure to set above temperature limit #2
Max_rpm_gadapt!	Fuel	Advanced	Maximum RPM global fuel injection adaption.
Max_vehicle_speed!	Misc	Basic	Maximum vehicle speed.
Min_dead!	Idle	Advanced	Rpm offset for lower integrator limit.
Min_load_gadapt!	Fuel	Advanced	Minimum Load global fuel injection adaption.
Min_rpm_gadapt!	Fuel	Advanced	Minimum RPM global fuel injection adaption.
Min_tid!	Fuel	Advanced	Lowest possible injectiontime in the system

Mis_temp_limit!	Diagnostics	Misfire	Below this cooling water temp limit no misfire are Resolution is 1.
Mis_trans_limit!	Diagnostics	Misfire	Disables misfire detection during throttle transients + 5 cumbustions. if (Trans_trott > Mis_trans_limit) Disable misfire detection Resolution is 1.
Mis1000_map!	Diagnostics	Misfire	Contains maximum allowed misfire over 2000 cumbustions for not causing emission level degradation. Resolution is 1.
Mis200_map!	Diagnostics	Misfire	Contains maximum allowed misfire over 400 cumbustions for not causing catalyst overheating. Resolution is 1.
Misfire_map_x_axis!	Misc	Axis	Control's the current working position in misfire map's depending on inlet manifold pressure. Resolution is 1.
Misfire_map_y_axis!	Misc	Axis	Control's the current working position in misfire map's depending on engine speed. Resolution is 10.
Open_all_varv!	Fuel	Advanced	When enginerev is coming under this limit the fuel will come back simultanously for all cylinders
Open_loop!	Sensor	Lambda sensor	Load limit where system switches to open loop. Function of rpm.
Open_loop_adapt!	Sensor	Lambda sensor	Open loop adaption, not used
Open_loop_knock!	Sensor	Lambda sensor	Load limit where system switches to open loop during knocking conditions. Function of rpm.
Open_varv!	Fuel	Advanced	When enginerev is coming under this limit the fuel will come back successively for all cylinders
Overs_tab!	Boost control	Advanced	Peek map, pressure peek (Overshoot)
Overs_tab_xaxis!	Boost control	Axis	Overshoot table x axis
Overstid_tab!	Boost control	Advanced	Overboost time
Ox1DeltaLoadLim!	Sensor	Lambda sensor	If Cat_delta_load exceeds this limit, oxygen sensor 1 transition test is halted.
Ox2_activity_time_lim!	Misc	Advanced	When Ox2_activity_time exceeds this limit the oxygen sensor 2 change activity is evaluated.
Ox2_change_lim!	Misc	Advanced	If Ox2_change is above this limit after Ox2_activity_time the oxygen sensor 2 activity test is approved.
P_fors!	Boost control	Advanced manual	Boost Regulation Map, P-Gain factor of PID control
P_fors_a!	Boost control	Advanced automatic	Boost Regulation Map, P-Gain factor of PID control (automatic)
P_man_korr_tab!	Misc	Advanced	Pressure correction vs. throttle
Pgm_mod!	Misc	Advanced	Program control flags
PMCal_CloseRamp!	Misc	Advanced	Ramp factor during close phase.
PMCal_IdlePosFiltCoef!	Misc	Advanced	Idle position filter coeficient. 1.000 -> Min filtering. 0.000 -> Max filtering.
PMCal_IdlePosIntABSLim!	Misc	Advanced	Halt limit for fast idle position changes.
PMCal_IdleRefHaltLim!	Misc	Advanced	Halt limit on idle reference position.
PMCal_IdleValueLim!	Misc	Advanced	Idle position change limit from reference value for a successful purge diagnostic test.
PMCal_LambdaAvgHaltLim!	Sensor	Lambda sensor	Halt limit on lambda average reference signal.
PMCal_LambdaAvgLim!	Sensor	Lambda sensor	Lambda change limit from reference value for a successful purge diagnostic test.
PMCal_LamTransCntLim!	Sensor	Lambda sensor	Number of transitions used for lambda integrator reference calculation.
PMCal_OpenPWMTab!	Misc	Advanced	PWM axis for purge monitor open phase.
PMCal_OpenTimeTab!	Misc	Advanced	Time axis for purge monitor open phase.
PMCal_RpmIdleNomRefLim!	Misc	Idle	RPM idle nominal change reference halt limit.
PMCal_StartRamp!	Misc	Advanced	Rampfactor to purge monitor start position (0%).
PMCal_TestCountLim!	Misc	Advanced	Number of tests that trionic will perform before purge test fails.
Press_trans_type!	Sensor	Basic	Saab part no of current pressure transducer.
Pulses_per_rev!	Runtime	Basic	Number of pulses from wheel during 1 rev.
Purge_tab!	Fuel	Advanced	The Purge matrix (0-100%)
Purge_temp!	Fuel	Advanced	The engine temperature above which Purge is alllowed.
Pwm_ind_rpm!	Misc	Axis	Rpm axis for several tables
Pwm_ind_trot!	Sensor	Advanced	Relative throttle value
Radius_of_roll!	Runtime	Basic	Front wheel tyre radius of roll.
Ramp_down_ram!	Ignition	Advanced	Ramp down table
Ramp_fak!	Fuel	Cranking	Correcting injectiontime with a faktor in this map at every halv enginerev when ramping during cranking.
Ramp_steg!	Sensor	Lambda sensor	Number of steps lambdaintergrator will change while ramping.
Ramp_up_ram!	Ignition	Advanced	Ramp up table

Reg_kon_fga!	Boost control	Advanced automatic	Regulation constant (automatic) below a programmable gear-ratio
Reg_kon_fgm!	Boost control	Advanced manual	Regulation constant for first and reverse gear manual
Reg_kon_mat!	Boost control	Advanced manual	Boost Regulation Map, BCV Constant (manual).
Reg_kon_mat_a!	Boost control	Advanced automatic	Boost Regulation Map, BCV Constant (automatic).
Reg_kon_sgm!	Boost control	Advanced manual	Regulation constant for second gear manual
Reg_last!	Boost control	Advanced	Boost regulation load axis values.
Reg_varv!	Boost control	Advanced	Boost regulation RPM axis values.
Regl_tryck_fgaut!	Boost control	Basic automatic	Maximum boost in first gear (automatic)
Regl_tryck_fgm!	Boost control	Basic manual	Maximum boost in first and reverse gear (limiter)
Regl_tryck_sgm!	Boost control	Basic manual	Maximum boost in second gear (limiter)
Restart_corr_hp!	Fuel	Cranking	Restart factor for heatplates
Restart_tab!	Fuel	Cranking	Restart factor (1 - 15, 15 - 30 and 30 - 45 seconds)
Restart_temp!	Fuel	Cranking	Restart temperature
Ret_delta_rpm!	Fuel	Enrichment	RPM drop enrichment, delta rpm for triggering
Ret_down_rpm!	Fuel	Enrichment	RPM drop enrichment, lower rpm limit for function
Ret_fuel_step!	Fuel	Enrichment	RPM drop enrichment, amount of decrement/combustion
Ret_fuel_tab!	Fuel	Enrichment	RPM drop enrichment, value for enrichment
Ret_fuel_time!	Fuel	Enrichment	RPM drop enrichment, constant enrichment time (ms)
Ret_up_rpm!	Fuel	Enrichment	RPM drop enrichment, upper rpm limit for function
Retard_konst!	Fuel	Transient	Fuel Injection Map, Deceleration, Throttle Position.
Retard_temp!	Fuel	Temperature compensation	Deceleration temperature compensation
Rev_grad_sens_cont!	Ignition	Advanced	Rev. gradient sensitivity continous
Rich_step_map!	Sensor	Lambda sensor	Rich step for lambda integrator
Rpm_max!	Misc	Basic	Maximum Revolutions Per Minute (RPM limiter)
Shut_off_time!	Misc	Advanced	Time between engine stop and disactivating main-relay
Sond_heat_tab!	Sensor	Lambda sensor	Load limit for lambda sensor heating
Sond_ign_limit!	Sensor	Lambda sensor	Test condition for front lambda sensor transition test.
Sond_omsl_limit!	Misc	Advanced	Threshold for transition test.
Start_dead_tab!	Fuel	Cranking	Number of dead injections.
Start_detekt_nr!	Misc	Advanced	The engine should be this number of halve enginerevs over the Start_detect_rpm to fulfil the second creterion of "Motorn_startad"
Start_detekt_rpm!	Misc	Advanced	Engine speed should be over this limit to fulfil the first creterion of "Motorn_startad"
Start_insp!	Fuel	Cranking	Base fuel amount (constant) while cranking
Start_max!	Fuel	Cranking	Number of halv enginerevs when injectiontime is constant
Start_proc!	Fuel	Cranking	Procentual adjusting of injection time (after Start_max) before ramping
Start_ramp!	Fuel	Cranking	Number of halv enginerevs to stop ramping and stabilize injectiontime.
Start_tab!	Idle	Advanced	Increment of valve position during start. Function of cooling water temperature)
Start_time_rpm_lim!	Fuel	Cranking	Rpm limit used for start time meassuring. Resolution is 1 rpm.
Start_trottel!	Fuel	Cranking	Throttlevalue which is needed to achieve total fuelcut while cranking
Start_v_vinkel!	Fuel	Cranking	Injection angle while cranking
Startvev_fak!	Fuel	Cranking	Enrichmentfactor on Start_insp depending on cooling water temperature
Synk_ok_limit!	Misc	Advanced	Syncronisation counter counting number of succesful sync detections.
Temp_max_regl!	Boost control	Advanced	Boost temperature limits
Temp_reduce_mat!	Fuel	Temperature compensation	Fuel Injection, Temperature compensation reduction map (open throttle)
Temp_reduce_mat_2!	Fuel	Temperature compensation	Fuel Injection, Temperature compensation reduction map (closed throttle)
Temp_reduce_x_st!	Fuel	Axis	Temperature reduction fuel correction x axis
Temp_reduce_y_st!	Fuel	Axis	Temperature reduction fuel correction y axis
Temp_steg!	Misc	Temperature calculation	Map-support points used in the most cooling water dependend maps

Tempkomp_konst!	Fuel	Temperature compensation	Temperature compensation
Tid_konst!	Fuel	Idle	When program goes with constant injection time during idle, the actual injection time is taken from this adress.
Trans_x_st!	Misc	Advanced	Delta throttle angle
Trans_y_st!	Misc	Advanced	RPM
Tryck_mat!	Boost control	Basic manual	Boost table for manual transmission
Tryck_mat_a!	Boost control	Basic automatic	Boost table for automatic transmission
Tryck_vakt_tab!	Boost control	Basic	Boost limit map (fuel cut)
Turbo_knock_tab!	Ignition	Advanced	Pressure limit for turbo knock regulation
Vinkel_konst!	Fuel	Advanced	If the program not interpolates delay-angle from crankshaftpulse to injectionstart in the Delay map, the angle will be taken from this constant.
Wait_count_tab!	Ignition	Advanced	Number of sparks to wait before reducing knock offsets

Appendix II : Trionic 5 pinout

70 pin connector

Pinnumber	Colour	Description	Symbol	Range	In/out
1	Red	+30 (power supply)		+12 Volt	In
2	Blue	Boost pressure control valve			Out
3	Grey	Injector 1			Out
4	Blue	Injector 2			Out
5	Yellow	Injector 3			Out
6	White	Injector 4			Out
8	---	Knock detect (knock indicator)		Switches to GND	Out
9	Brown/white	Trigger 1			Out
10	Green	Trigger 2			Out
11	Blue	Trigger 3			Out
12	Grey	Trigger 4			Out
13	White	Engine torque limitation			In
14	Orange	Drive			In
15	White	Brake light switch			In
17	Yellow	Combustion cylinder 1+2			In
18	Brown	Combustion cylinder 3+4			In
20	PK				
21	BU	M96: EVAP canister purge valve diagnosis M96.5: Fuel tank pressure sensor		0-5 Volt	In
22	Grey	MAP sensor input signal	P_Manifold	0 – 5 Volt	In
23	Green	Oxygen sensor 1 (front lambda sonde)	AD_sond		In
24	Black/white	Main ground			In
25	Black/white	Main ground			In
26	Green	Boost pressure control valve			Out
27	Yellow/red	EVAP canister purge valve			Out
28	GN	EVAP canister close valve			Out
29	---	Heat plate activation relay			Out
31	Yellow/white	Main relay			Out
32	White	Check engine			Out
33	Blue/white	Diagnosis (data link under seat, TECHII)			In/Out
35	YE/BK	Engine load signal out			Out
36	BU/WH	Cruise control			In
38	YE	Low fuel signal light			In
39	Green	Speed signal	Bil_hast		In
41	Yellow	Crankshaft position sensor			In
42	WH/BK	Throttle position sensor			Out
43	Blue/white	MAP (Manifold Absolute Pressure) signal		+5 Volt	Out
44	Orange	Knock signal	AD_knock		In
45	GN/BN	Throttle position sensor	AD_trot		In
46	Black	Intake air temperature sensor	AD_lufttemp		In
47	Black	Oxygen sensor, reference ground			In
48	Red	+30 (power supply)		+12 Volt	In
49	BU/VT	Idle air control valve (IAC)			Out
50	Black	Oxygen sensor 1, preheating			Out
51	BK/WH	Oxygen sensor 2 (rear), preheating			Out
52	YE/WH	Vacuum pump, brake system			Out
54	RD/WH	A/C out			Out
55	Brown	Shift up light (to instrument cluster)		Switches to GND	Out
56	Violet	Fuel pump relay			Out
57	Yellow	Throttle position signal			Out
58	Blue	Engine speed signal (RPM)	Rpm		Out
59	Green/white	A/C in			In
60	Green/white	+15 (switched power supply, ignition contact)			In
62	Grey	CANH (CAN bus high signal)			In/Out
63	Red	CANL (CAN bus low signal)			In/Out
65	Red/white	Programming voltage		+16 Volt	In
66	Black	Reference ground for engine coolant temp. sensor		0 Volt	In
67	Black/white	Sensor ground		0 Volt	Out
68	Yellow	Engine coolant temperature sensor	AD_kylv		In
69	---	Exhaust gas temperature (not mounted)	AD_EGR	0-5 volt	In
70	---	Oxygen sensor 2 (rear lambda sonde)	AD_cat		In

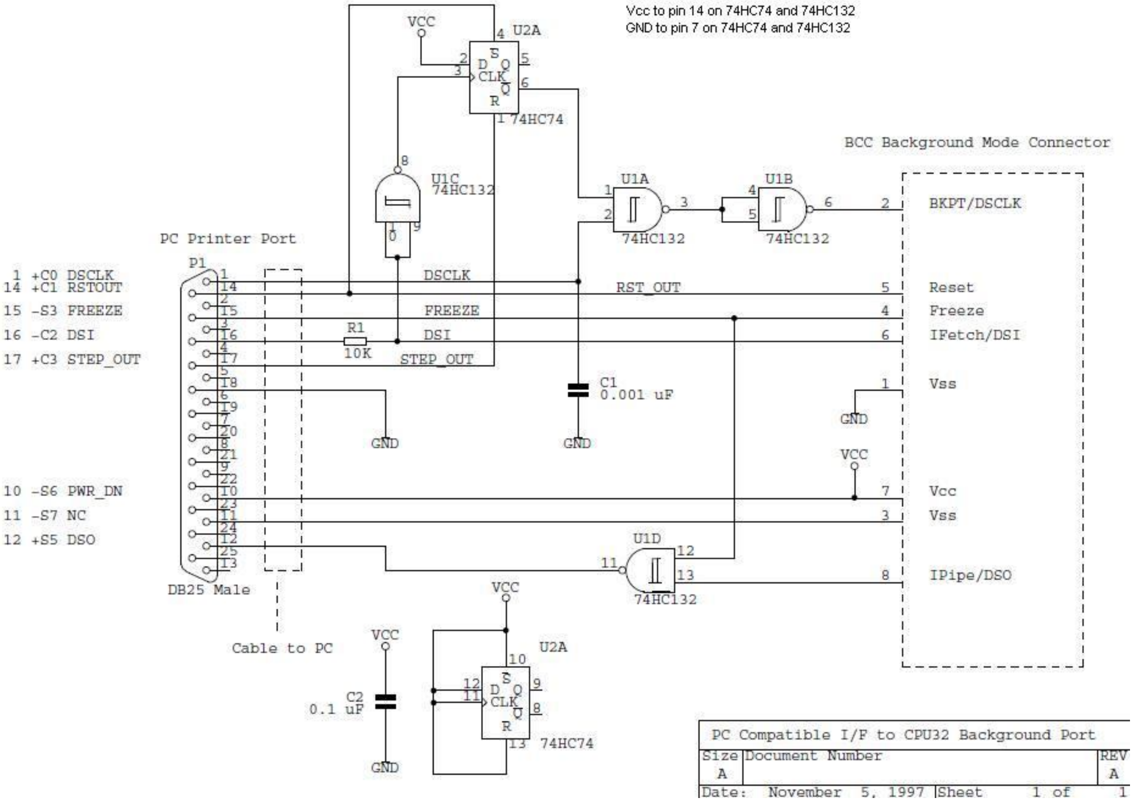
Appendix III : BDM technical information

General

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://trionic.mobixs.eu/bd32-122.zip.

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.



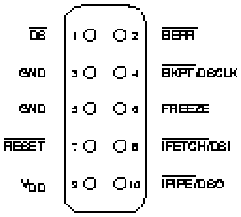
The table below shows the component list that you need to build the interface. Of course a soldering iron, PCB etc. are things that you also need.

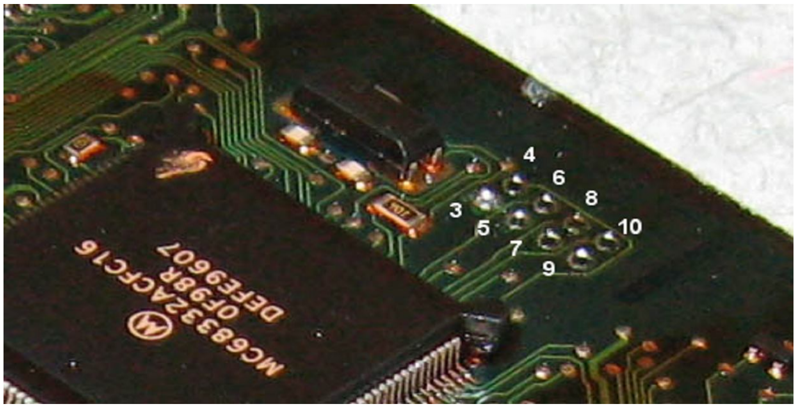
Component	Amount	Description
74HC74	1	Dual JK Flip-Flop with Set and Reset
74HC132	1	Quad 2-input NAND Schmitt Trigger
Capacitor 0.1 uF	1	

Capacitor 0.001 uF	1	
Resistor 10 kΩ	1	
LPT cable + connector	1 meter	Smart is to get PCB type with normal LPT cable.
10 wire flatcable	20 cm	
10 pin female header for flatcable	1	

Pinout

Pinnumber	Pin name	Description	Remark
1	DS	Data strobe from target MCU. Not used in current interface circuitry	Not used with Trionic 5
2	BERR	Bus error input to target. Allows development system to force bus error when target MCU accesses invalid memory	Not used with Trionic 5
3	VSS	Ground reference from target	Look at image below
4	BKPT/DSCLK	Breakpoint input to target in normal mode; development serial clock in BDM.Must be held low on rising edge of reset to enable BDM	Look at image below
5	VSS	Ground reference from target	Look at image below
6	FREEZE	Freeze signal from target. High level indicates that target is in BDM	Look at image below
7	RESET	Reset signal to/from target. Must be held low to force hardware reset	Look at image below
8	IFETCH/DSI	Used to track instruction pipe in normal mode. Serial data input to target MCU in BDM	Look at image below
9	VCC	+5V supply from target.BDM interface circuit draws power from this supply and also monitors 'target powered/not powered' status	Look at image below
10	IPIPE/DSO	Tracks instruction pipe in normal mode. Serial data output from target MCU in BDM	Look at image below




4
6
8
3
5
10
7
9
MC68332ACFC16
0F98R
DEFE9607

Appendix IV : 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 5 cars there are 2 commonly used turbo chargers: the Garrett T25 for B204E, B204S, B204L, B234E and B234L engines and the Mitsubishi TD04-HL-15G/T (6cm2) for the B234R engines.

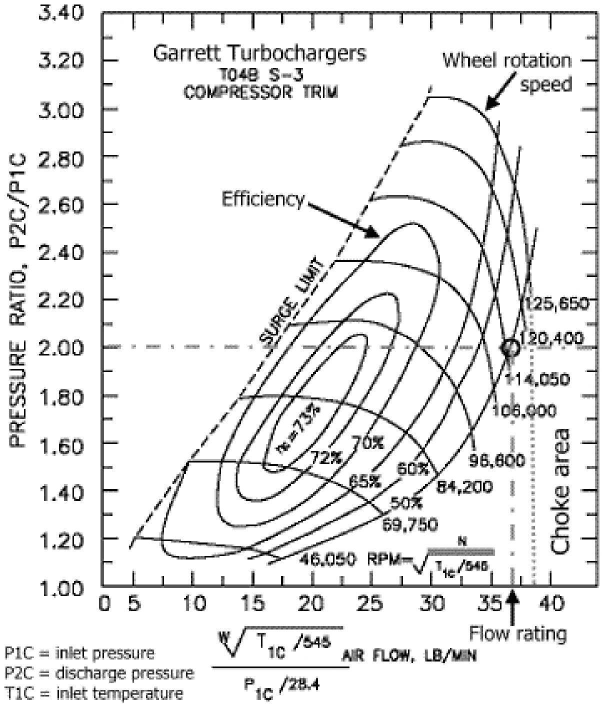
http://www.automotivearticles.com/Turbo_Selection.shtml

Terms to know:

- 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)
  Example:
  Air Temperature (Air Temp) - 60°F
  Barometric Pressure (Baro) - 14.7 psi
  Engine air consumption (Actual Flow) = 50 lb/min
  $$\text{Corrected Flow} = \frac{\text{Actual Flow}\ \sqrt{([\text{Air Temp}+460]/545)}}{\text{Baro}/13.95}$$
  $$\text{Corrected Flow} = \frac{50*\sqrt{([60+460]/545)}}{14.7/13.95} = 46.3\ \text{lb/min}$$
- Pressure Ratio
  Ratio of absolute outlet pressure divided by absolute inlet pressure
  Example:
  Intake manifold pressure (Boost) = 12 psi
  Pressure drop, intercooler ($\Delta P_{Intercooler}$) = 2 psi
  Pressure drop, air filter ($\Delta P_{Air\ Filter}$) = 0.5 psi
  Atmosphere (Atmos) = 14.7 psi at sea level
  $$PR = \frac{Boost + \Delta P_{Intercooler} + Atmos}{Atmos - \Delta P_{Air\ Filter}}$$
  $$PR = \frac{12+2+14.7}{14.7-.5} = 2.02$$

How to read compressor maps

A 2-dimensional pressure map looks like this.



Turbo compressor map

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.

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 sounic 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.

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 litre motor with this turbo needs 15psi of manifold pressure to flow a certain CFM. A 3 litre 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.

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.

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.

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.

Understanding information within the compressor map

1. The oblong ovals on the chart or “islands” as they are called represent the efficiency of the turbo in that range. As you can see on this map, the most efficient operation (73%) is in the very centre of the chart. This is general characteristic of most turbochargers. Without getting into the thermodynamics of adiabatic heat-pumps, we’ll just say that efficiency is a measure of how much excess heat the turbo puts into the compressed air coming out of the outlet. So intuitively, more efficient is better.

2. Wheel rotational speed is simply the rpm at which the compressor wheel is spinning.

3. The choke point, which is usually not indicated on flow maps, is the maximum flow rating the turbo is capable of regardless of pressure or efficiency.

4. Beyond the surge limit on the left of the plot, compressor surge occurs. In laymen’s terms, this phenomenon is caused by a back pressure wave entering the exit of the compressor housing and disrupting flow through the compressor wheel. Surge will kill turbos and is to be avoided at all costs.

Surge limit

To the left of the surge limit line on the flow map is the *surge area* where compressor operation can be unstable. Typically, surge occurs after the throttle plate is closed while the turbocharger is spinning rapidly and the by-pass valve does not release the sudden increase in pressure due to the backed-up air. During surge, the back-pressure build-up at the discharge opening of the compressor reduces the air flow. If the air flow falls below a certain point, the compressor wheel (the impeller) will loose its "grip" on the air. Consequently, the air in the compressor stops being propelled forward by the impeller and is simply spinning around with the wheel, which is still being rotated by the exhaust gas passing through the turbine section. When this happens, the pressure build-up at the discharge opening forces air back through the impeller causing a reversal of air flow through the compressor. As the back pressure eventually decreases, the impeller again begins to function properly and air flows out of the compressor in the correct direction. This sudden air-flow reversal in the compressor can occur several times and may be heard as a repetitive "Whew Whew Whew" noise if the surge is mild (such as when the by-pass valve is set a little too tight) to a loud banging noise when surge is severe. Surge should be prevented at all costs because it not only slows the turbocharger wheels so that they must be spooled back up again but because it can be very damaging to the bushings or bearings and seals in the center section.

Selecting a different turbo charger

Calculating your engine’s flow requirements

Now that you can read and understand a compressor flow map, its time to figure out how to match a turbo to your engine, this involves selecting the proper compressor and turbine wheels along with the right combination of housing A/R. A mismatched turbo could not only result is extreme lag, but also wasted potential as a turbo can easily outflow an engine. I.e. bigger is not always better.

The only real calculation that needs to be done is to determine how much air you engine is actually flowing. This depends on a number of things including the RPM, absolute temperature (Rankin, equal to 460 + Fahrenheit temp), absolute manifold pressure (psi, equal to boost pressure plus atmospheric pressure), and lastly the engine volumetric flow or EVF in cfm.

First to calculate EVF use the following equation:

$$EVF = \left(\frac{engineCID}{1728}\right) * \left(\frac{RPM}{2}\right)$$

engineCID = Engine displacement in cubic inches.
Next we’ll use EVF to calculate the amount of air in lb/min the engine is flowing under boost and at temperature using this equation:

$$N = \frac{P * EVF * 29}{10.73 * T}$$

Where N is the airflow in lb/min, P is the absolute pressure in psi, and T is the absolute ambient temperature in Rankin.

Finally, multiply N by the volumetric efficiency of your engine (VE). This compensates for the fact that upon every cycle of the engine, not all of the old air/fuel mix in the cylinders is forced out the exhaust. Thus there is a difference between the actual airflow through and engine and the predicted airflow. This discrepancy is equated to a VE. There is literally thousands of hours worth of online reading about volumetric efficiencies for just about every production engine. To get the most accurate results from this step I would suggest researching your engine and coming up with the most realistic VE possible as this does have a significant affect on engine flow. If you are just messing around with compressor flow maps and need a value for VE just to experiment with, 85% efficiency is a nice conservative number for most modified turbocharged cars at high rpm (6500-7500). Keep in mind though that on a forced induction setup VE can easily exceed 100% so again it will be very beneficial to research *your* engine.
For our SAAB engines these number apply.

EVF / RPM	2.0L engine (122 cubic inch)	2.3L engine (140 cubic inch)
1000 rpm	35.3	40.6
2000 rpm	70.6	81.2
3000 rpm	105.9	121.8
4000 rpm	141.3	162.4
5000 rpm	176.6	203.1
6000 rpm	211.9	243.7
7000 rpm	247.2	284.3

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 20°C. 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 boostlevel.
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.

Airflow / boostpressure	2.0L lb/min	2.3L lb/min
0.2 bar (2.9 psi)	15.4	17.7
0.4 bar (5.8 psi)	18.0	20.6
0.6 bar (8.7 psi)	20.5	23.6
0.8 bar (11.6 psi)	23.1	26.5
1.0 bar (14.5 psi)	25.7	29.5
1.2 bar (17.4 psi)	28.2	32.4
1.4 bar (20.3 psi)	30.8	35.4
1.6 bar (23.2 psi)	33.4	38.3

In T5Suite 2.0 you can see your maximum boost request curve plotted in the compressor map of your choice.

Determining the Best Wheel Trim-Housing A/R Combination

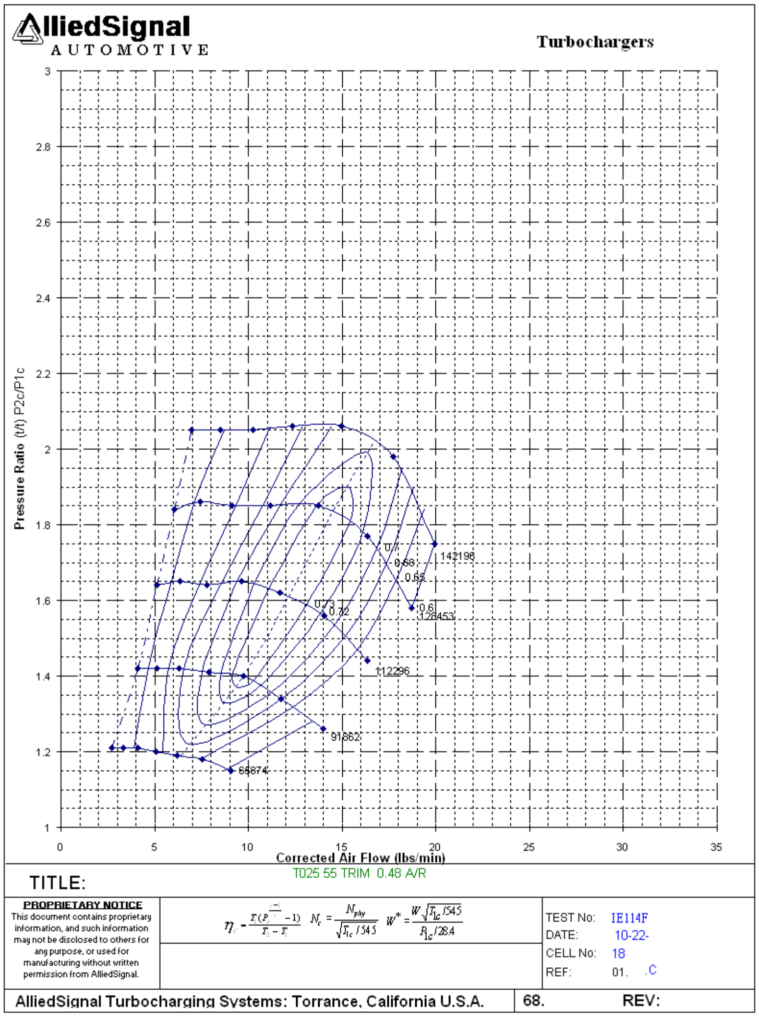
With the flow rate you have just calculated, you can look at compressor maps of different turbo chargers to see which ones give you the air flow you need at the pressures and efficiencies that you want to run.

When selecting a turbo, it is important to do the above calculations for a number of different RPM's and boost pressures because you will not always be at redline under full boost while driving you car. Checking the turbo performance at various engine speeds and pressures will give the overall picture of how well the turbo is sized to your vehicle.

Matching a flow map to your engine flow requirements will allow you to pick the compressor wheel trim for your application. However before you can go out and purchase that new turbo, you still have to settle on an exhaust wheel and turbine A/R. The real determining factor in this selection is maintaining compressor wheel speed. Remember the wheel RPM lines on the flow map? Well a properly sized exhaust wheel/housing combination will keep the compressor wheel operating within the maximum and minimum wheel speeds on the map as often as possible. Since different "hot side" combinations can affect your turbo's performance, (i.e. a little more lag in return for more top end, or quicker spool up at the cost of overall power) the best thing to do is to contact a turbo manufacturer or distributor (www.extremeturbo.com, www.forcedpeformance.com, www.turbochargers.com) and they will be able to tell you the exact effects you can expect from all of the various hot side combos available for your turbo model.

Garrett T25 specifications

Weight: 7,5kg
Compressor diameter: 60mm
Turbine diameter: 65mm and 59mm
Bearing span: 37,8mm
Moment of inertia: 5,4x10-5 kg/m2
Oil flow: 1,7L/min / SAE30 / 90C / 2,75bar
Compressor wheel: 54mm 55 trim, A/R 48
Turbine wheel: 53.8mm 61 trim, A/R 49



TITLE: T025 55 TRIM 0.48 A/R

PROPRIETARY NOTICE
This document contains proprietary information, and such information may not be disclosed to others for any purpose, or used for manufacturing without written permission from AlliedSignal.

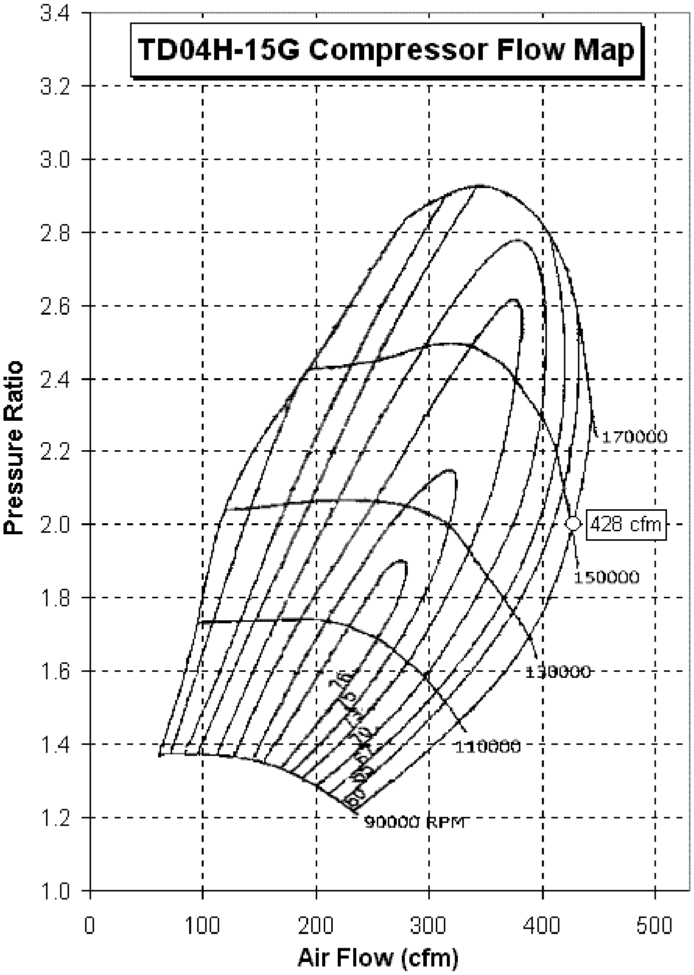
$\eta_c = \frac{T_1(P_r^{\frac{\gamma-1}{\gamma}}-1)}{T_2-T_1}$ $N_c = \frac{N_{phy}}{\sqrt{T_{1c}/545}}$ $W^* = \frac{W\sqrt{T_{1c}/545}}{P_{1c}/28.4}$

TEST No: IE114F
DATE: 10-22-
CELL No: 18
REF: 01. .C

AlliedSignal Turbocharging Systems: Torrance, California U.S.A. 68. REV:

As you can see the T25 (trim 55) can flow 18 lbs of air per minute @ pressure ratio = 2 and efficiency will be ~65%. This 18 lbs/minute converts to about 260 cfm. The maximum efficiency zone (73%) reaches up to pressure ratio 1.9. This would be ~0.9 bar overpressure.
In terms of usage the T25 can take us up to ~1.1 bar boost pressure and bring up to 250 bhp.

Mitsubishi TD04-15G specifications

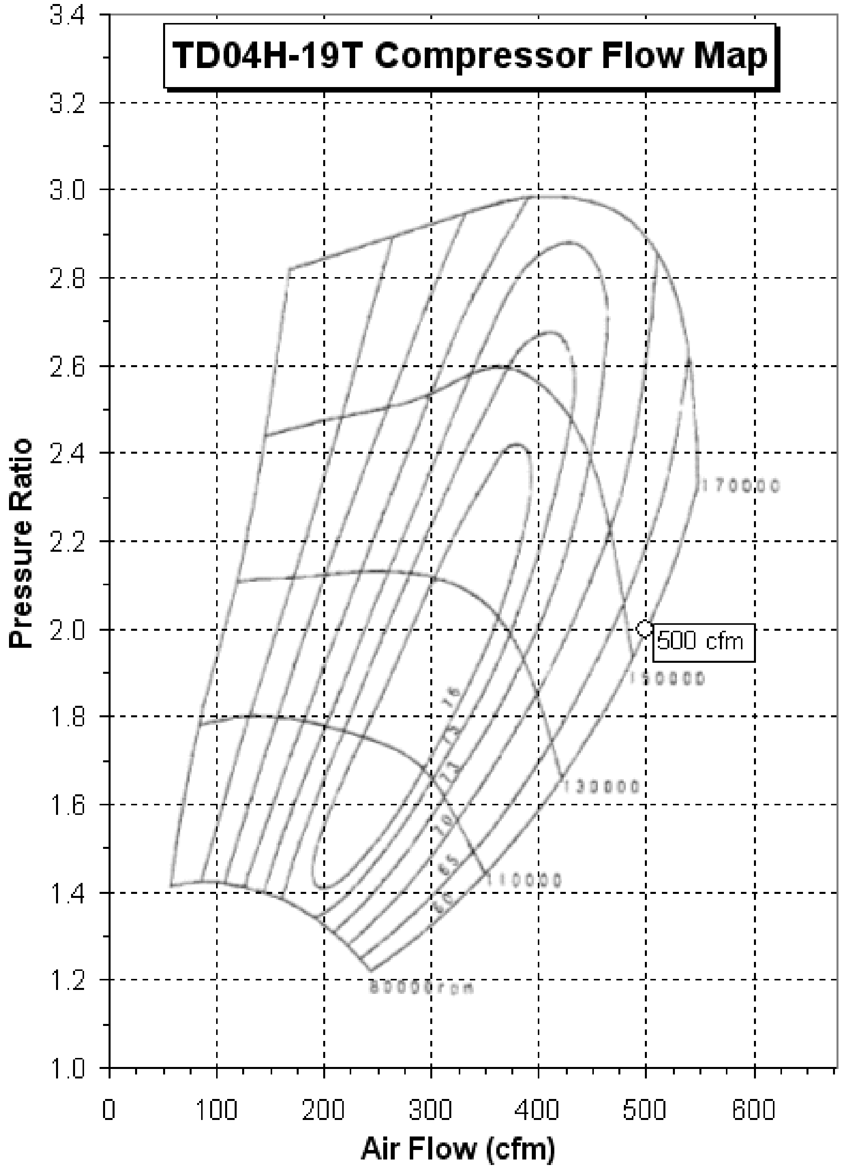


The TD04-15G from the 9000 Aero can flow more air than the "little" T25. Looking at the map we can see that the TD04-15G can flow ~428 cfm @ pressure ratio 2 and efficiency of 60%.

Also, the maximum efficiency zone (76% vs. 73% for the T25) reaches up to 1.9 pressure ratio with would be ~0.9 bar.

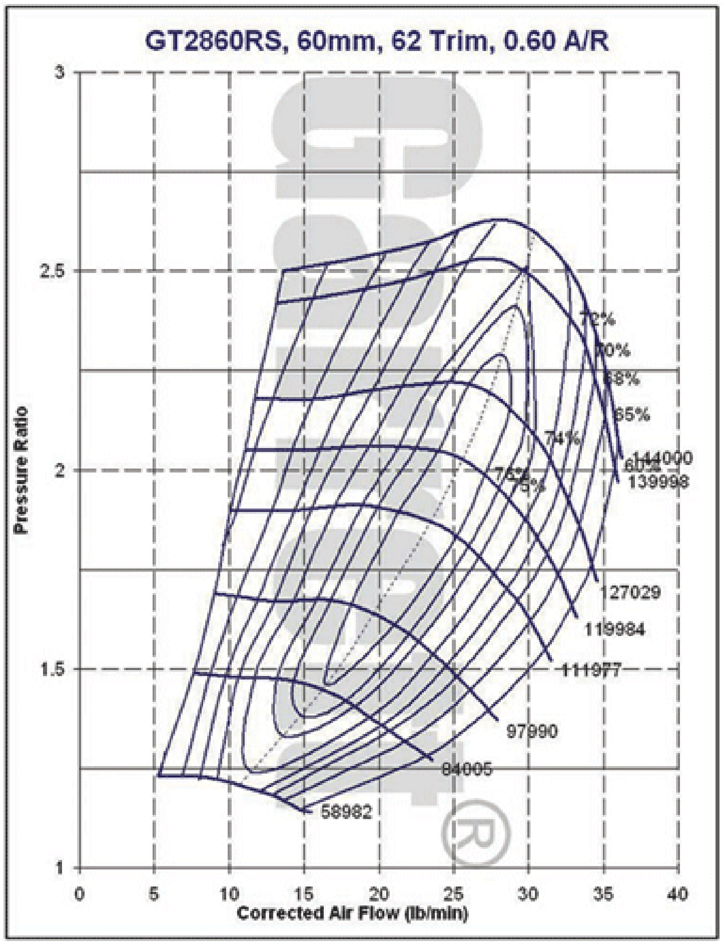
In terms of usage the TD04 can take us up to ~1.4 bar boost pressure. The TD04 can bring up to ~330bhp.

Mitsubishi TD04-19T specifications



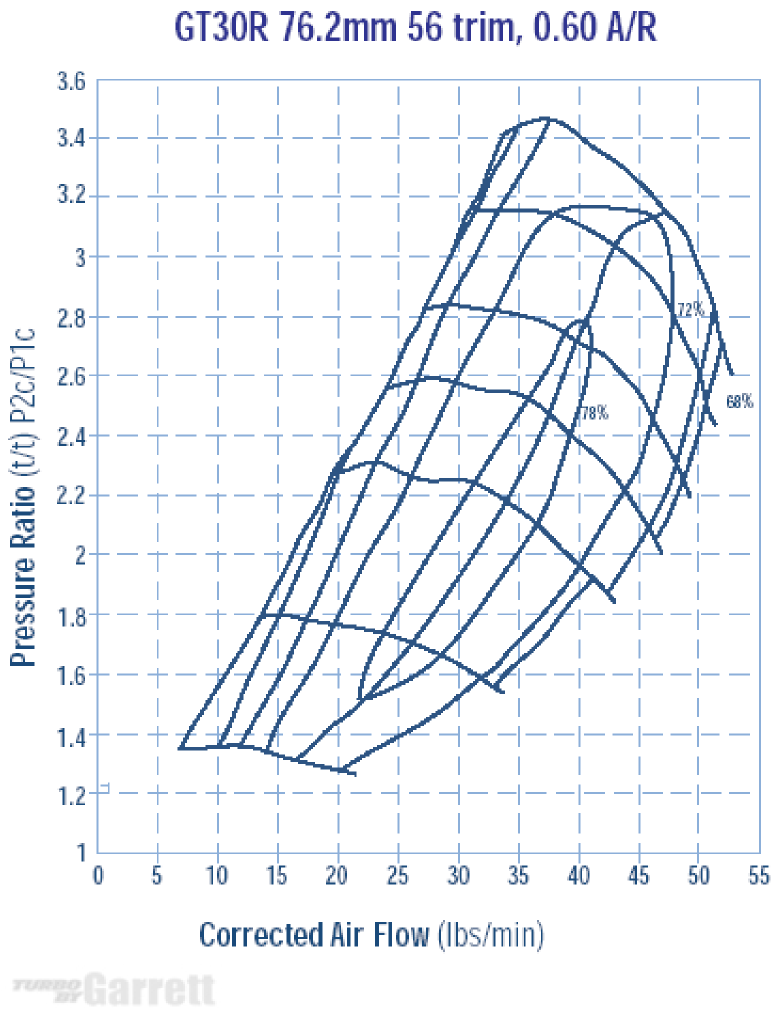
This TD04-19T compressor map has the air flow axis noted in m3/s. 1 m3 per second means 2118 cfm. So this turbo can flow approximately 0.26 m3/s = 550 cfm @ pressure ratio 2.3.
At the standard ratio of 2 it still flows 510 cfm which is more than a GT28RS. The TD04-19T can bring up to ~380 bhp.

Garrett GT28RS (GT2860R) specifications



The GT28RS is frequently used in modified cars. It can flow much more air than the TD04 and the T25. If we look at the compressor map for the GT28RS (black lines) we see that this turbo is capable of flowing ~33 lbs/minute which is ~477 cfm at pressure ratio 2. The maximum efficiency runs much higher than that. Up to 1.4 bar boost pressure this turbo will run at 75% efficiency.
In terms of usage the GT28RS can run nicely up to 1.6 bar at an air flowrate of ~37 lbs/min = 535 cfm. This is much more than the TD04-15G of course. The GT28RS can bring up to ~350 bhp.

Garrett GT30R specifications



The GT30R (GT3071) is given for reference reasons only. It can flow even more air than the GT28RS. If we look at the compressor map we see that this turbo is capable of flowing ~45 lbs/minute which is ~650 cfm at pressure ratio 2. The maximum efficiency runs up to 1.8 bar boost pressure. In terms of usage the GT30R can run nicely up to 1.8 bar at an air flowrate of ~52 lbs/min = 750 cfm. Even within the maximum efficiency zone this turbo will flow 40 lbs/minute which is more than the GT28RS will flow even at maximum.
The GT30R can bring up to ~500 bhp.

Conclusion

Comparing the two compressor maps for TD04 and T25 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 V: Upgrade stages 1-7

If you want to go beyond the standard stages I – III there's more to alter than just the "silly" stuff like ECU, exhaust and catalyst. This chapter will describe what steps are needed for stages 1 up to 7.

Stage I

The target amount of power for stage I is about 235 bhp for FPT versions (T25 turbo) and 260 bhp for Aero models (TD04-15 turbo)

Component	Stock	Stage I minimum requirement
ECU	200/225 bhp	Stage I
Exhaust	2"	2"
Intake	---	---
Catalyst	---	---
Injectors	345 cc/min	345 cc/min
Fuel lines	---	---
Turbo	T25/TD04-15	T25/TD04-15
Exhaust manifold	---	---
Intercooler	---	---
Clutch	450 Nm	450 Nm
Camshafts	---	---
Fuel pump	---	---
Wastegate	---	---
Mapsensor	2.5 bar	2.5 bar
Air delivery pipe	---	---
Cylinder head	---	---

Stage II

The target amount of power for stage III is about 250 bhp for T25 models and 270 bhp for TD04 models.

Component	Stock	Stage II minimum requirement
ECU	200/225 bhp	Stage II
Exhaust	2"	3" cat-back
Intake	---	---
Catalyst	---	---
Injectors	345 cc/min	345 cc/min
Fuel lines	---	---
Turbo	T25/TD04-15	T25/TD04-15
Exhaust manifold	---	---
Intercooler	---	---
Clutch	450 Nm	450 Nm
Camshafts	---	---
Fuel pump	---	---
Wastegate	---	---
Mapsensor	2.5 bar	2.5 bar
Air delivery pipe	---	---
Cylinder head	---	---

Stage III

The target amount of power for stage III is about 270 bhp for T25 models and 280 bhp for TD-04 models.

Component	Stock	Stage III minimum requirement
ECU	200/225 bhp	Stage III
Exhaust	2"	3" turbo back
Intake	---	Open/sport air filter
Catalyst	---	Sport (3") catalyst
Injectors	345 cc/min	345 cc/min
Fuel lines	---	---
Turbo	T25/TD04-15	T25/TD04-15
Exhaust manifold	---	---
Intercooler	---	---
Clutch	450 Nm	450 Nm
Camshafts	---	---
Fuel pump	---	---
Wastegate	---	---
Mapsensor	2.5 bar	2.5 bar
Air delivery pipe	---	---
Cylinder head	---	---

Stage IV

The target amount of power for stage IV is about 300 bhp. From stage 4 there's no longer a difference between FPT and Aero models because the T25 turbo cannot reach a stage IV level and has to be replaced from this stage on.

Component	Stock	Stage IV minimum requirement
ECU	200/225 bhp	Stage IV
Exhaust	2"	3" turbo back
Intake	---	Open/sport air filter
Catalyst	---	Sport (3") catalyst
Injectors	345 cc/min	413 cc/min
Fuel lines	---	---
Turbo	T25/TD04-15	TD04-15
Exhaust manifold	---	---
Intercooler	---	---
Clutch	450 Nm	Sports clutch (600 Nm)
Camshafts	---	---
Fuel pump	---	---
Wastegate	---	Reinforced model (Forge)
Mapsensor	2.5 bar	3 bar
Air delivery pipe	---	---
Cylinder head	---	---

Stage V

The target amount of power for stage V is about 350 bhp

Component	Stock	Stage V minimum requirement
ECU	200/225 bhp	Stage V
Exhaust	2"	3" turbo back
Intake	---	Open/sport air filter
Catalyst	---	Sport (3") catalyst
Injectors	345 cc/min	413 cc/min
Fuel lines	---	---
Turbo	T25/TD04	TD04-18T/19T
Exhaust manifold	---	---
Intercooler	---	Cross flow, high capacity
Clutch	450 Nm	Sports clutch (600 Nm)
Camshafts	---	---
Fuel pump	---	---
Wastegate	---	Reinforced model (Forge)
Mapsensor	2.5 bar	3 bar
Air delivery pipe	---	---
Cylinder head	---	---

Stage VI

The target amount of power for stage VI is about 400 bhp

Component	Stock	Stage VI minimum requirement
ECU	200/225 bhp	Stage VI
Exhaust	2"	3" turbo back
Intake	---	Open/sport air filter
Catalyst	---	Sport (3") catalyst
Injectors	345 cc/min	630 cc/min
Fuel lines	---	---
Turbo	T25/TD04	Garrett GTBB30
Exhaust manifold	---	---
Intercooler	---	Cross flow, high capacity
Clutch	450 Nm	Sports clutch (600 Nm)
Camshafts	---	Upgraded model for better engine breathing
Fuel pump	---	Walbro 255 or Bosch 044
Wastegate	---	Reinforced model (Forge)
Mapsensor	2.5 bar	3 bar
Air delivery pipe	---	---
Cylinder head	---	---

Stage VII

The target amount of power for stage VII is about 500 bhp

Component	Stock	Stage VII minimum requirement
ECU	200/225 bhp	Stage VII
Exhaust	2"	3" turbo back
Intake	---	Open/sport air filter
Catalyst	---	Sport (3") catalyst
Injectors	345 cc/min	750 cc/min
Fuel lines	---	Upgraded piping and rail
Turbo	T25/TD04	Garrett GT30R
Exhaust manifold	---	Tubular model with all mandrels of same length
Intercooler	---	Cross flow, high capacity
Clutch	450 Nm	Sports clutch (700 Nm)
Camshafts	---	Upgraded model for better engine breathing
Fuel pump	---	Walbro 255 or Bosch 044
Wastegate	---	Reinforced model (Forge)
Mapsensor	2.5 bar	3 bar
Air delivery pipe	---	Upgraded model (Abbott)
Cylinder head	---	Custom ported

Appendix VI: Check Engine Light (CEL)

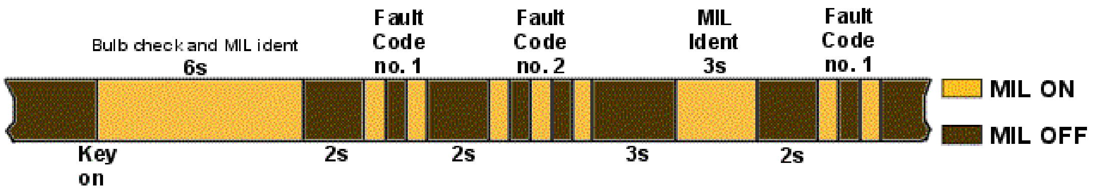
If the Trionic detects something is wrong it may light the CEL. The CEL flashes its code when you turn on the ignition and wait for a few seconds. If more than one code it in the fault list, the codes will be flashed one after another. The sequence is repeated until you turn off the ignition or start the car.

CEL Flash codes

In the table below you will find the known CEL flash codes with their description.

# flashes	Description	Additional information
2	MAP sensor	Pressure sensor out of boundaries or defective
3	Intake air temperature sensor	IAT sensor out of boundaries or defective
4	Coolant temperature sensor	Coolant water sensor defective
5	Throttle position sensor	TPS out of boundaries or defective
6	Oxygen sensor (lambda)	Oxygen sensor out of boundaries or defective
7	Air/Fuel mixture (A/F ratio)	A/F ratio out of boundaries (outside of adaption range)
8	EVAP value	
9	ECU internal failure	Flash / SRAM fault or checksum error.

# flashes	Description	ISAT codes
2	MAP sensor	P0105 P0106 P0107 P0108
3	Intake air temperature sensor	P0110 P0112 P0113
4	Coolant temperature sensor	P0115 P0117 P0118
5	Throttle position sensor	P0120 P0121 P0122 P0123
6	Oxygen sensor (lambda)	P0130 P0135
7	Air/Fuel mixture (A/F ratio)	P0170 P0171 P0172
8	EVAP value	P0443 P1443 P1444 P1445
9	ECU internal failure	P1651 P1652 (check ground, connectors and checksum)



Appendix VII: Knock and misfire detection

Trionic detects knocking and misfires by means of the ionization current that flows between the spark plug gaps. This appendix will explain how this works.

http://www.fs.isy.liu.se/~larer/Projects/main.html

Ionization current generation

In an ideal combustion reaction, hydrocarbon molecules react with oxygen and generate only carbon dioxide and water, e.g. isooctane gives

$$C_8H_{18} + \frac{25}{2}O_2 \longrightarrow 8CO_2 + 9H_2O.$$

In the combustion there are also other reactions present, that include ions, which go through several steps before they are completed.

$$\begin{aligned} CH + O &\longrightarrow CHO^+ + e^- \\ CHO^+ + H_2O &\longrightarrow H_3O^+ + CO \\ \\ CH + C_2H_2 &\longrightarrow C_3H_3^+ + e \quad . \end{aligned}$$

These ions, and several others, are generated by the chemical reactions in the flame front. Additional ions are created when the temperature increases as the pressure rises.

The processes creating the ionization current are complex and are also varying from engine cycle to engine cycle. *Image 25* shows ten (10) consecutive cycles of the cylinder pressure and the ionization current operating at constant speed and load.

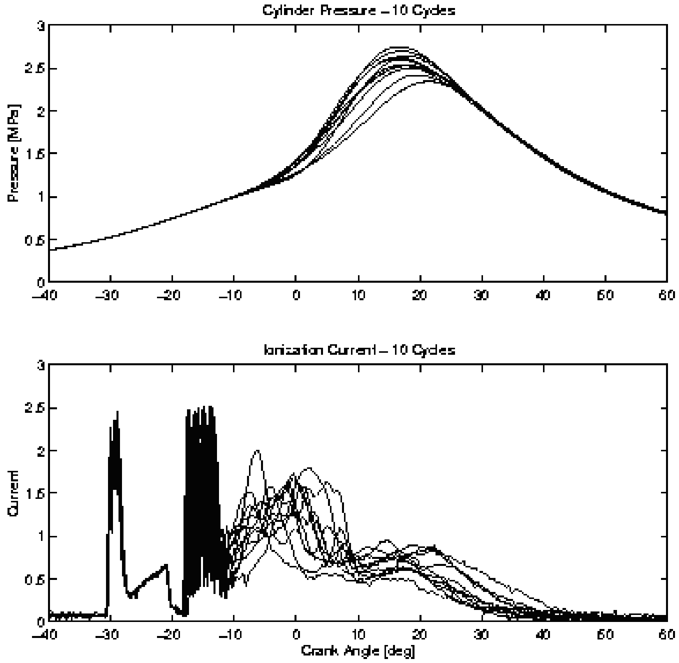


Image 25: Cycle to cycle variations in the combustion

As can be seen, the cycle-by-cycle variations are significant, which is a given problem in interpreting these signals.

Ionization current sensing

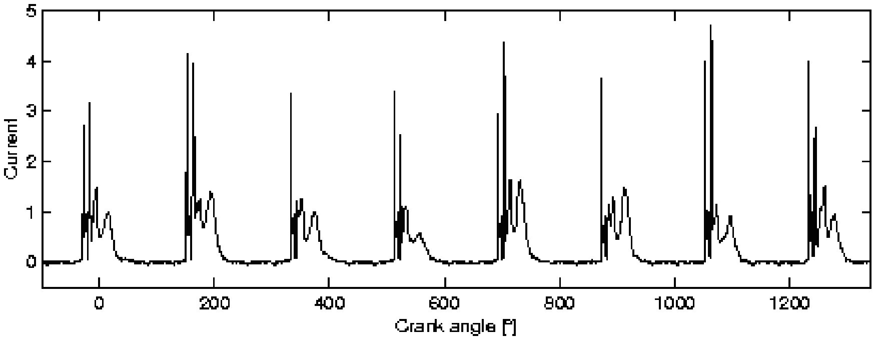


Image 26: Ionization current in one cylinder

Knock is a pressure oscillation in the cylinder with a frequency determined by the geometry of the combustion chamber. The oscillation is present in the current measurement and can be extracted mainly by using a band pass filter in a well chosen time window of the current signal. Knocking can destroy the engine. When there is a misfire, then there are no resulting ions and hence no current which is easily detected. Misfires can and will destroy the catalyst.
Ionization current interpretation can be used for both purposes, knock detection and misfire detection.

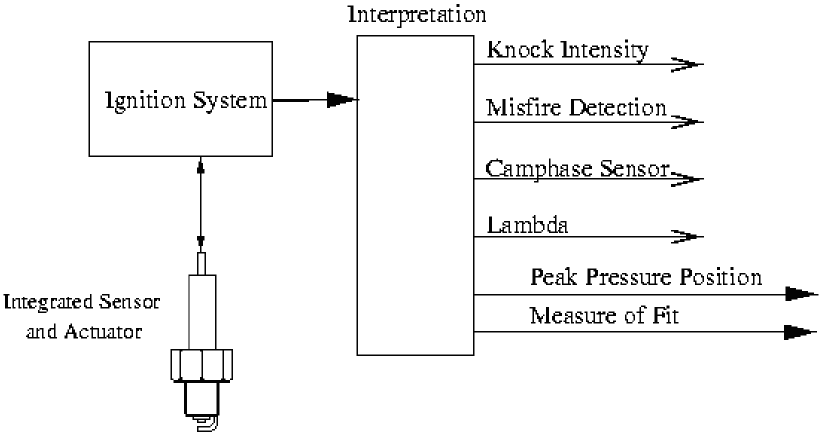


Image 27: Possible sensor information from ionization current

Detection

To detect the ions, a DC bias is applied to the spark plug, generating an electrical field. The electrical field makes the ions move and generates an ion current. A schematic illustration is shown in *Image 28 (a)*. The current is measured at the low-voltage side of the ignition coil, and does not require protection from the high-voltage pulses in the ignition, *Image 28 (b)*.

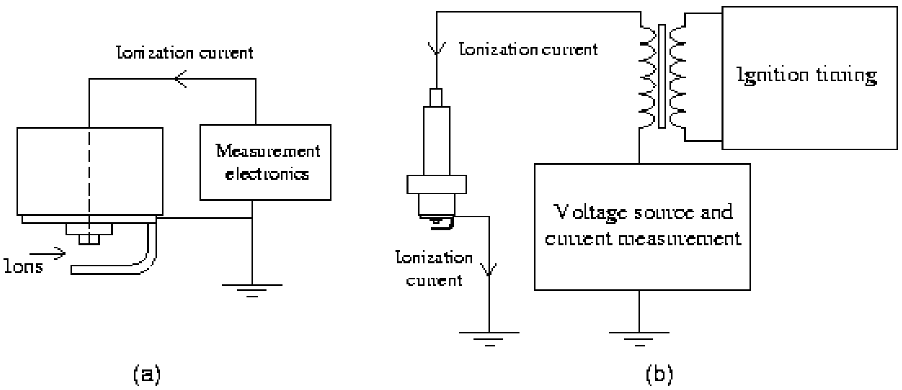


Image 28: Measuring the ionization current

Measurement of the ionization current. (a) The spark plug-gap is used as a probe. (b) Measurement on the low voltage side of the ignition coil.

The ionization current is an interesting engine parameter to study. It is a direct measure of the combustion result that contains a lot of information about the combustion, and several challenges remain in the interpretation of it. Some of the parameters that affect the ionization current are: temperature, air/fuel ratio, time since combustion, exhaust gas recycling (EGR), fuel composition, engine load, and several others.

Ionization Current Terminology

The ionization current typically has three phases: a phase related to ignition, a phase related to ions from the flame development and propagation, and a phase related to pressure and temperature development. In *Image 29*, the three phases of the ionization current are displayed. Each of these phases has varying characteristics and they also mix together in complicated ways. In the ignition phase, the ionization current is large, with reversed polarity. Due to the high current in the ignition the measured signal shown in the figure is limited. What can be seen in the image too is the ringing phenomenon in the coil after the ignition.

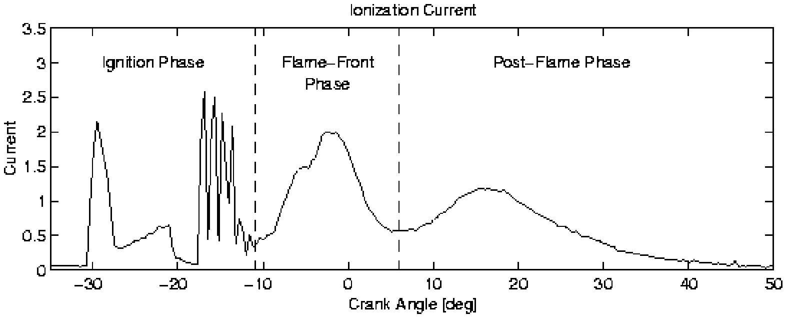


Image 29: Ionization current phases: ignition, flame front, and post flame

In the flame-front phase, the high level of ions associated with the chemical reactions in the flame produces one or more characteristic peaks. The ions generated by the flame have different recombination rates. Some ions recombine very quickly to more-stable molecules, while others have longer residual times. The result is a high peak which after some time decays as the ions recombine.

In the post-flame phase the most stable ions remain, generating a signal that follows the cylinder pressure due to its effect on the temperature and molecule concentration. Ions are created by the combination of the measurement voltage and the high temperature of the burned gases, since the temperature follows the pressure during the compression and expansion of the burned gases, when the flame propagates outwards and the combustion completes. The ionization current thus depends on the pressure.

Spark Advance and Cylinder Pressure

The spark advance is used to position pressure development in the cylinder such that the combustion produces maximum work. Under normal driving conditions the mixture is ignited around 15 – 30° in crank angle before the piston has reached top dead center (TDC), and the pressure peak comes around 20 degrees after TDC. In the graph below three different pressure traces, resulting from three different spark timings, are shown. Earlier spark advance normally gives higher maximum pressures and maximum temperatures that appear at earlier crank angles.

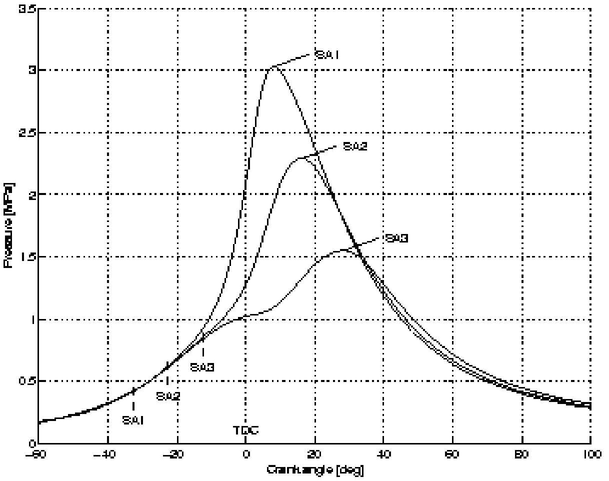


Image 30: Cylinder pressure vs. Ignition timing

Three different pressure traces resulting from three different spark advances. The different spark advances are; **SA1**: spark advance 32.5° before top dead center (TDC), **SA2**: 22.5° before TDC, **SA3**: 12.5° before TDC. The optimal spark advance is close to **SA2**.

The optimal spark advance for maximum output torque is close to **SA2** for the operating point in the figure, and the resulting peak pressure position lies around 17° after TDC. With too early ignition timing the pressure rise starts too early and counteracts the piston movement. This can be seen for the pressure trace with spark advance **SA1** where the pressure rise starts already at -20° due to the combustion. There are also losses due to heat and crevice flow from the gas to the combustion chamber walls, and with an earlier spark advance the loss mechanisms start earlier reducing the work produced by the gas. Higher pressures give higher temperatures which also decrease the difference in internal energy between the reactants and products in the combustion, thus resulting in lower energy-conversion ratios. The heat loss mechanisms and the lower conversion ratio can be seen in *Image 30*, at crank angles over 30°, where the pressure trace from the **SA1** spark advance is lower than the others.

Too late ignition gives a pressure increase that comes too late so that work is lost during the expansion phase. In *Image 30*, the pressure increase for spark advance **SA3** starts as late as at TDC. But work is also gained due to the later start of the effects mentioned above, which also can be seen in the figure. The pressure trace from the spark advance, **SA3**, is higher than the others at crank angles over 30°. However, this gain in produced work can not compensate for the losses early in the expansion phase.

Peak Pressure Concept

Thus, optimal spark advance positions the pressure trace in a way that compromise between the effects mentioned above. To define the position of the in-cylinder pressure relative to TDC, the peak pressure position (PPP) is used, *Image 31*. The PPP is the position in crank angle where the in-cylinder pressure takes its maximal value. There also exist other ways of describing the positioning of the combustion relative to crank angle, e.g. based on the mass fraction burned curve.

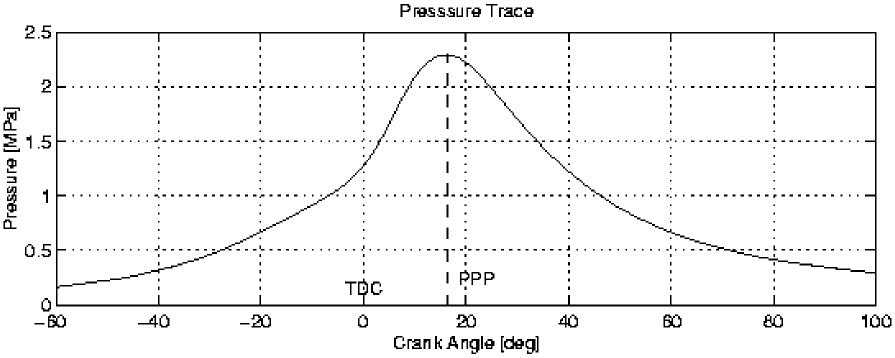


Image 31: Peak pressure position

Engine-tuning for efficiency

To be able to get the maximum torque from the engine at a given load point we have to investigate the torque development in different settings. In *Image 32*, mean values, over 200 cycles, of the PPP are plotted together with the mean value of the produced torque at four different operating points covering a large part of the road load operating range for the engine. Two of the operating points have an engine speed of 1500 rpm with different throttle angles, and for the two other operating points the engine speed is doubled to 3000 rpm. The PPP for maximum output torque in the figure lies around 15°ATDC (after TDC) for all these operating points, , even though the spark advance differs a lot.

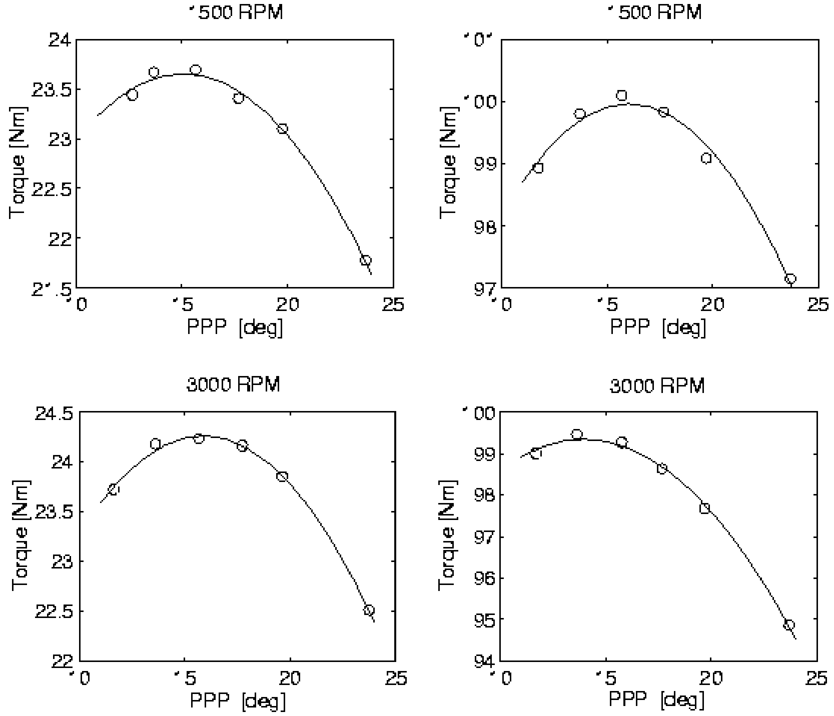


Image 32: PPP vs. torque in different settings

Note that the load and speed are changed over large intervals, and that the PPP for maximum output torque at the different operating points does not differ much. The PPP versus torque curve is flat around the position for the maximum. Therefore a spark schedule that maintains a constant PPP at 15° is close to optimum. Considering only the work produced, this motivates that an optimal spark schedule maintains almost the same position for the peak pressure. However, the optimal PPP changes slightly with the operating points. The efficiency can thus be improved a little bit further by mapping the optimal PPP for each operating point, and provide these values as reference signal to the spark timing controller. The peak pressure positioning principle can also be used for meeting emission standards.

Appendix VIII: Sensors and actuators

This appendix will list details about the sensors and actuators used in a T5 car.

General

Sensors are devices used to gather information. In a Trionic 5 car a lot of sensors are used to determine what actions to take inside the ECU. These sensors are all analogue, which means they output a signal that has to be converted to digital numbers for the ECU to be able to understand them. Actuators on the other hand are devices that enable the ECU to interact with the processes in the car. Actuators are driven (or activated) by the Trionic be it directly or indirectly.

Sensors

The range of sensors used by Trionic is impressive. This chapter will list all used sensors with their characteristics ranging from temperature sensors to pressure sensors and crankshaft sensors.

Intake Air Temperature (IAT) sensor

The intake air temperature sensor is needed to determine the density of the air entering the system. The density of the air determines the amount of O_2 molecules in a cubic inch of air and the amount of O_2 molecules determine how much fuel should be injected into the cylinder to get a stoichiometric burn. Hence, fuel injection duration depends on IAT. The IAT sensor outputs a analogue signal between 0 and 5 volts as shown in the table below. The sensor is located on the air intake tube leading to the throttle body. Through the holes in the sensor head, the intake air flows past the NTC (negative temperature coefficient) resistor in the sensor. The ECU applies a voltage across the sensor. The voltage drop across the sensor drops as temperature rises. Based on this temperature and the MAP input the air mass can be determined continuously by the ECU. Pinnumber 1 of the sensor (Black wire) connects to ground while pinnumber 2 (white wire), which carries the output voltage connects to pinnumber 46 on the ECU. The value for actual intake air temperature can be monitored through CAN bus using symbol *Lufttemp.*

°Celcius	°Fahrenheit	Voltage	Resistance Ω
-30	-22	4.5	20-30
-10	+14	3.9	8.3-10.6
+20	+68	3.2	2.3-2.7
+40	+104	1.5	1.0-1.3
+60	+140	0.9	0.56-0.67
+80	+176	0.7	0.30-0.36

Manifold Air Temperature (MAT) sensor

TODO: Insert description for MAT here.

Manifold Absolute Pressure (MAP) sensor

For continuous measurement of the pressure in the intake manifold a pressure sensor is present. It is mounted on the front of the bulkhead compartment and it is connected to the intake manifold by a vacuum hose. The MAP sensor is a piezo electrical element which reacts to pressure induced to it. The element converts the pressure to a linear voltage on its output lead in which low pressure results in a low voltage and high pressure results in a high voltage. This output voltage ranges from 0 to 5 volts. The sensor has an internal circuit to compensate for temperature fluctuations. Based on this pressure and the air temperature in the intake manifold the air mass can be determined continuously by the ECU. The mapsensor needs a supply voltage, ground and has a output voltage and thus has 3 leads on it. Pinnumber 1 (black wire) connects to ground, pinnumber 2 (blue/white wire) carries the pressure voltage signal and is connected to pinnumber 22 on the ECU while pinnumber 3 (grey wire) connects to pinnumber 43 on the ECU which supplies the sensor with +5 volts. The table below shows the output voltages for different pressures.

Pressure (bar)	Output voltage
-0.75	0.4
-0.50	0.9
0.00	1.8
+0.25	2.3
+0.5	2.8
+0.75	3.3
+1.00	3.8
+1.25	4.3

The MAP sensor is a 2.5 bar type. This would theoretically mean that it can measure up to ~1.5 bar boost pressure. In practice however the sensor saturates before that. This saturation takes place at about 1.4 bar which would be ~4.75 volts. If you plan to achieve more boost than 1.4 bar you will have to use a 3 bar MAP sensor which can measure correctly up to ~1.9 bar. See chapter "Converting to three bar mapsensor" for more information on this subject. The value for actual pressure in the manifold can be monitored through CAN bus using symbol *P_Manifold* or *P_Manifold10.*

Coolant Temperature sensor

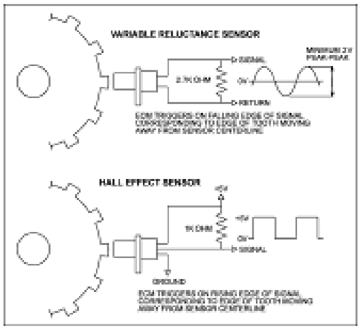
`

Trionic uses the coolant temperature sensor to determine the engine's temperature. Engine temperature indicates whether – for example – warmup ignition map should be used or what compensation to make for injection (warmup enrichtment) and so on. The coolant temperature sensor signal is applied to pinnumber 68 on the ECU. The value for actual coolant temperature can be monitored through CAN bus using symbol *Kyl_temp.*

Crankshaft position sensor

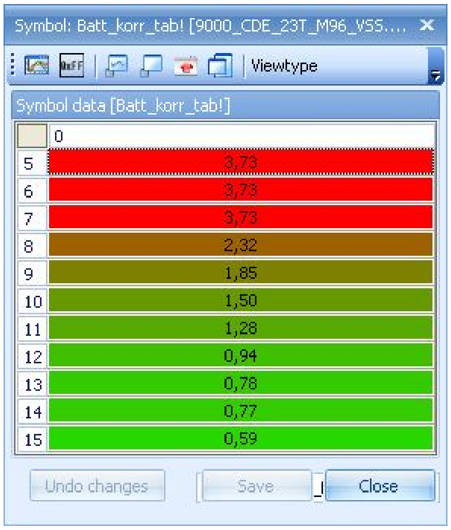
⚠ Trionic T5.2 uses a different crankshaft position sensor than Trionic T5.5.
The early (T5.2) models use a HALL effect sensor while the later models (T5.5) uses a variable reluctance sensor.

The HALL effect sensor produces a square wave output while the VR type produces a sine wave output as shown in the picture below. The VR type output voltage is converted by T5.5 by using the LM1815. This results in a similar output as the HALL effect sensor. T5 has 62 pulses per engine rotation from which 2 teeth are missing at 117° BTDC on cylinder 1 to determine TDC. The crankshaft sensor signal is applied to pinnumber 41 on the ECU. This sensor also supplies the ECU with engine speed information. The amount of pulses per second indicates how many rpm the engine is making. The crankshaft signal is applied to pinnumber 41 on the ECU. The RPM signal is applied to pin 58.


VARIABLE RELUCTANCE SENSOR
MINIMUM 2V PEAK-PEAK
SIGNAL
2.7K OHM
0V
RETURN
ECM TRIGGERS ON FALLING EDGE OF SIGNAL CORRESPONDING TO EDGE OF TOOTH MOVING AWAY FROM SENSOR CENTERLINE
HALL EFFECT SENSOR
+5V
1K OHM
+5V
0V
SIGNAL
GROUND
ECM TRIGGERS ON RISING EDGE OF SIGNAL CORRESPONDING TO EDGE OF TOOTH MOVING AWAY FROM SENSOR CENTERLINE

Battery voltage

Trionic uses the battery voltage as an input to compensate for lag in the opening time of the fuel injectors. Lower battery voltage means that the injectors will open slower and hence injector open time needs to be longer to get the same amount of fuel injected into the cylinders. The table below shows the battery voltage correction table.



Symbol data [Batt_korr_tab!]

	0
5	3,73
6	3,73
7	3,73
8	2,32
9	1,85
10	1,50
11	1,28
12	0,94
13	0,78
14	0,77
15	0,59

Throttle Position sensor (TPS)

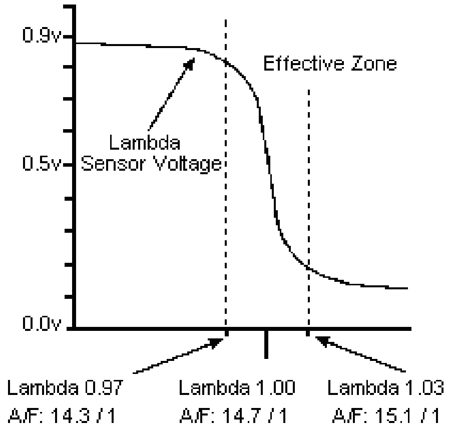
The throttle position sensor is used by Trionic to regulate charging pressure, ignition timing and for supplying a richer mixture during acceleration and a leaner mixture during deceleration. The maximum charging pressure can only be obtained at wide open throttle (WOT). The sensor is different for cars with and cars without TCS (traction control system).
Cars without TCS obtain the TPS signal from pin 3 of the throttle position sensor and it is applied to pin 45 of the ECU. Signal voltages range from 0.2 volt while idling up to 4 volts at wide open throttle. Cars with TCS obtain the TPS signal from pin 26 of the ETS module (this is a PWM signal) and it is applied to pin 57 of the ECU. Signal voltages range from 0.2 volt while idling up to 4 volts at wide open throttle. The average throttle position can be monitored through CAN bus through symbol *Medeltrot*.

% throttle plate open	Output voltage
0	0.20
10	0.40
25	0.95
50	2.10
75	2.85
90	3.40

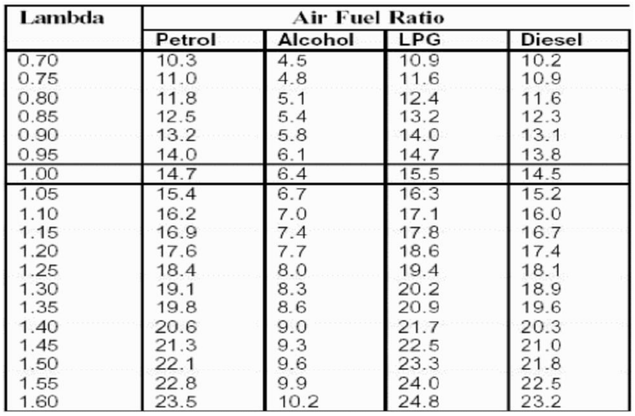
100	4.00

Lambda sonde (oxygen sensor)

To be able to measure the result of the combustions in terms of it being stoichiometric Trionic gets input from one or two (depending on make year and market) oxygen sensors a.k.a. lambda sensors or sondes. These sensors output a voltage that indicates the amount of air (O_2) left in the exhaust gases. T5 uses narrowband sensors which means that the output voltage they supply is not very accurate. The image below shows that the effective zone for a narrowband sensor is from ~14 to 15 AFR. The Trionic get a signal < 0.1 volt from the sensor the mixture is too lean while a signal > 0.9 volt means that the mixture is too rich. Normally the output voltage will swing from one side to the other and thus keeping the average mixture result correct for the catalyst. The front lambda sensors output voltage is applied to pinnumber 23 on the ECU.



Please note that different fuels need different mixtures to obtain lambda 1.



Lambda	Air Fuel Ratio			
	Petrol	Alcohol	LPG	Diesel
0.70	10.3	4.5	10.9	10.2
0.75	11.0	4.8	11.6	10.9
0.80	11.8	5.1	12.4	11.6
0.85	12.5	5.4	13.2	12.3
0.90	13.2	5.8	14.0	13.1
0.95	14.0	6.1	14.7	13.8
1.00	14.7	6.4	15.5	14.5
1.05	15.4	6.7	16.3	15.2
1.10	16.2	7.0	17.1	16.0
1.15	16.9	7.4	17.8	16.7
1.20	17.6	7.7	18.6	17.4
1.25	18.4	8.0	19.4	18.1
1.30	19.1	8.3	20.2	18.9
1.35	19.8	8.6	20.9	19.6
1.40	20.6	9.0	21.7	20.3
1.45	21.3	9.3	22.5	21.0
1.50	22.1	9.6	23.3	21.8
1.55	22.8	9.9	24.0	22.5
1.60	23.5	10.2	24.8	23.2

Knock sensor

As mentioned in appendix VII, Trionic uses the spark plugs via the direct ignition cassette to sense knocking and misfires. Please refer to this appendix for more information. The knock signal is applied to pinnumber 44 on the ECU.

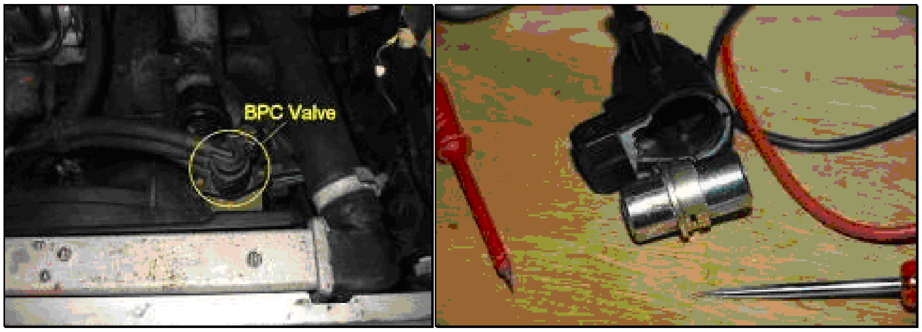
Vehicle speed signal

The ECU needs the vehicle speed signal to determine in which gear we are driving. This signal is a square wave form which varies in frequency depending on vehicle speed. The signal is applied to pinnumber 39 on the ECU.

Actuators

Boost control valve

To be able to control the charging pressure in the turbo, Trionic utilizes a pneumatic valve to pressurize the wastegate actuator. The system actually uses charging pressure from the turbo to pressurize the actuator so no charging pressure automatically means no control over the wastegate by Trionic. This is not a problem because if there is no charge pressure no overboost can occur anyway. The charging pressure is led to the input of the boost control valve and normally – when no interaction from Trionic occurs – actuates the wastegate so the valve opens. Normally, Trionic keeps the wastegate shut by activating the valve. If Trionic decides it is time to open the wastegate (be it a little bit or much) it shifts the valve so that part of the charge pressure is applied to the waste actuator. This is its turn opens the wastegate valve and lets some of the exhaust gases escape the turbo – to the exhaust – without it spinning the turbine wheel. This way, the amount of air driving the turbine wheel decreases and of course the compressor wheel spins slower. As a result the charging pressure decreases as well. The images below show the position of the valve and the internals. The ECU controls the BPC through pins 2 and 26 on the ECU. The value for output duty cycle on the valve can be monitored through CAN bus using symbol *PWM_ut10.*



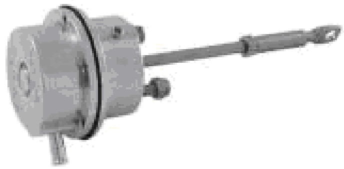
Wastegate actuator

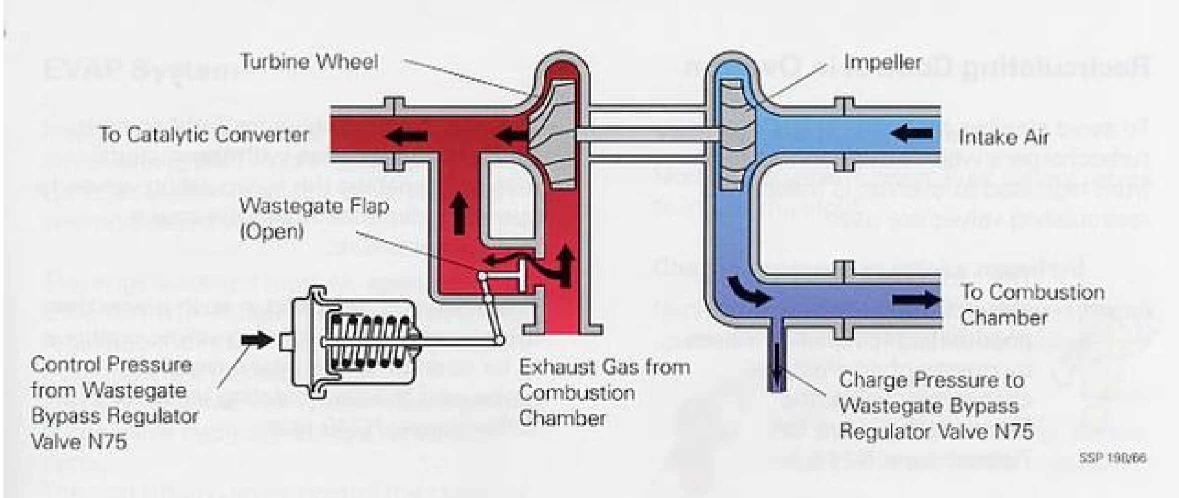
The wastegate actuator is controlled by the airflow from the BPC (see above) and is the physical connection to the wastegate valve. More airpressure on the actuator means that the wastegate valve opens more. The wastegate valve is kept shut by a spring that ensures that the valve closes when the air pressure is no longer applied. Sometimes overshoot problems with charging pressure are due to a worn / weak wastegate spring. The pictures below show the wastegate actuator and a diagram showing the working of the actuator. The higher the duty cycle (DC) on the solenoid valve, the more air pressure bleeds off to the compressor inlet side. If the turbo is not making pressure - e.g. compressor is atm - and the solenoid is not being activated (2% DC) there is no pressure on the actuator and hence the spring in the actuator is not being pushed back (counter pressured). The wastegate will remain shut by the pressure of the spring.
If the turbo is making boost - say 1.5 bar over atm - and the DC = 2%, the pressure will also be pumped into the wastegate actuator and it will counter act the pressure (tension) of the spring. The gate will open. If there's a high DC (98%) on the valve, the majority of the boost pressure from the compressor side will be leaked to the compressor inlet side by the valve and nearly no pressure will arrive in the wastegate actuator. The valve will be kept shut by the spring tension.
An exception on this is when the pressure drop OVER the turbo is higher than the spring tension in which case the gate will be pushed open from the outside (the exhaust side). In this case back pressure is too high. Because the gate will be pushed open, boost pressure will drop and the gate will close again. Boost pressure will rise again and the cycle will start over again resulting in fluctuating

boost.

Disconnecting the W-hose results in 'unlimited' boost because the compressor will not be able to pressurize the actuator and it will therefore keep the gate shut because of the spring tension.

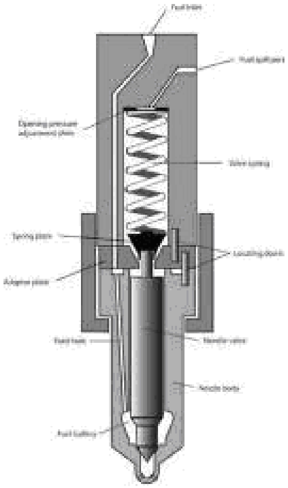
So: low DC means the ECU is not controlling boost pressure OR it tries to lower boost by pressurizing the wastegate actuator.
high DC means the ECU is trying to raise or maintain a certain boost level.
If you see a maximum DC (98%) and boost pressure still doesn't reach the requested level you'll have a spring that is too weak or you have too much back pressure pushing the gate open from the outside (which can also mean a spring that is too weak but also an exhaust that is too restrictive).





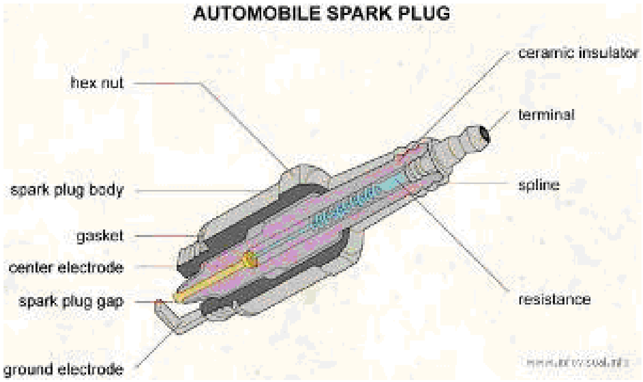
Injectors

The fuel injectors enable the Trionic to inject fuel in neatly dosed amounts into each cylinder. The injectors are actually small solenoid valves that are controlled by the ECU. The pressure over the injector is kept stable, normally at three bar. Opening the injector for a certain amount of time will inject a known amount of fuel into the cylinder. See chapter "Converting T5 to larger injectors" for more information. The image below shows how a injector is build up. The injectors are triggered by pinnumbers 3, 4, 5, and 6 on the ECU for - respectively - injector 1, 2, 3, and 4. The value for actual injection duration can be monitored through CAN bus using symbol *Insp_tid10.*



Spark plugs

The spark plugs enable the Trionic to ignite the fuel mixture inside the combustion chamber. In a T5 car the sparkplugs are indirectly driven by the ECU which means that the ECU signals the direct ignition cassette to fire a certain spark plug. See chapter "Ignition" for more information. The image below shows the internals of a spark plug. The ECU fires the plugs through pins 9, 10, 11 and 12 for respectively plug 1, 2, 3 and 4. The value for actual ignition angle can be monitored through CAN bus using symbol *Ign_angle.*



DI Cartridge

The Direct Ignition cartridge (DI cassette) had the following pinout on its connector.

Pinnumber	Description	ECU pinnumber
1	Used for monitoring the charge voltage to the capacitor (T7)	
2	Trigger signal cylinder 1	9
3	Trigger signal cylinder 2	10
4	Trigger signal cylinder 3	11
5	Trigger signal cylinder 4	12
6	Ground	
7	Knock sensor function	44
8	Combustion signal cylinders 1+2	17
9	Combustion signal cylinders 3+4	18
10	Main power from relay	

Cylinder number 4 is the one closest to the connector. Pin number 1 is used for monitoring the charge voltage to the capacitor (1/100 of the charge voltage gives about 0-4V). The voltage on the trigger pins is 12V and it triggers on the falling flank when grounded (500mA peak). The trigger must at least be held low for the discharge time of the coil which is about 500µs. There must be a 2.5ms charge time between triggers to avoid damage to the cassette. The recharge of the capacitor starts directly when the trigger goes high again and the charge time is 2.5ms at 12V battery voltage. It is important to minimize voltage drop in the harness as the charge time will rapidly increase with lower voltage (3ms at 9V). Charge voltage can be monitored the on pin 1.

The misfire signal is a square wave signal 0-12V. There are 2 lines for this, one is for cylinders 1&2 and the other is for cylinders 3&4. SAAB also uses this signal during cranking to find which cylinder is in the compression stroke and which is in the exhaust stroke. Trionic triggers both partner cylinders until it can detect a combustion in one of the cylinders. During this time these two coils get half each of the stored energy in the Cap. You will have a signal some microseconds after a trigger if you have a combustion. This signal is only for combustion detect and has nothing to do with knock.

The raw ION knock signal is filtered in the DI cassette to a 7kHz +/- 6V sine wave. The amplitude of the signal reflects the degree of knock. The charge pulse gives a lot of disturbance so the trigger must be kept low until the knock info is read. Normally most of this information is found between 10-40º ATDC. If the spark is 30º BTC and the knock info is at 40º ATDC, this means the trigger signal should be 70º long. Trionic has a "window" for when reading out the knock which varies in time depending on MAP and RPM. Knock window timing can be found in Knock_wind_on and Knock_wind_off tables.

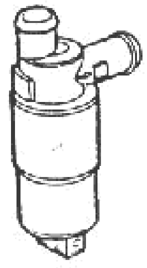
There are several different DI cassettes, one red and one black for four cylinder engines. They have been in production since 1987. The black ones have AC spark, giving longer spark duration. This can improve idle stability and lean condition misfire. There is also a three cylinder cassette, which can be found in 9000 & 900 models with V6 engine.

Purge valve (carbon canister solenoid valve)

The carbon canister contains active charcoal or active carbon granules. Most evaporation control systems reduce the emission of fuel vapour during the time the vehicle is idling in traffic, or parked in strong sunshine by absorbing the vapour fumes into the carbon canister. Once the engine is at its normal operating temperature the stored hydrocarbons are released into the inlet manifold where they become part of the combustible air/fuel mixture. The control for allowing the hydrocarbons to be released into the inlet manifold through a cut off valve is achieved electrically. The electronic solenoid is controlled by the Electronic Control Module (ECU) by switching the earth path to ground under specified conditions. The purge valve is controlled by pinnumber 27 and 28 on the ECU.

Idle Air Control (IAC) valve

The engine idling speed on Trionic 5 cars is controlled by an electrically operated idle air control (IAC) valve, also referred to as an automatic idle control (IAC) valve on some models. One end of this valve is connected by a rubber hose to the air filter side of the throttle plate, while the other end is connected by a rubber hose to the engine side of the plate. By varying the opening of the valve, the engine management system can vary the amount of air (and fuel) supplied and maintain an appropriate idle speed regardless of engine load, for example from the air conditioning system. The picture below shows the IAC valve. The valve is controlled by pin 49 on the ECU.



Appendix IX: Converting to T7 boost control valve

Short description

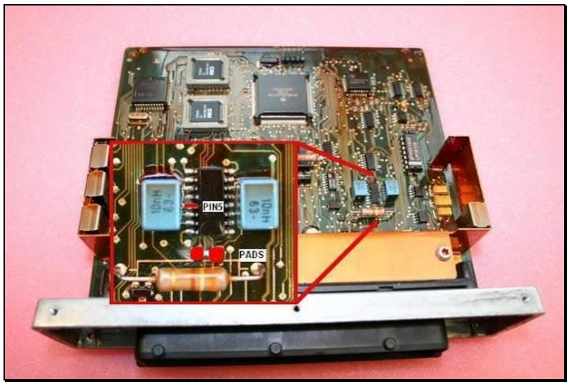
This appendix describes the steps necessary to use the Trionic 7 BPC (solenoid valve that controls the wastegate actuator) on a Trionic 5 car with a Trionic 5 ECU. Normally this is not possible and that is a shame because the valves for T5 are much more expensive and prone to failure simply because of their construction. The changes needed for this conversion consist of some software and some hardware changes.

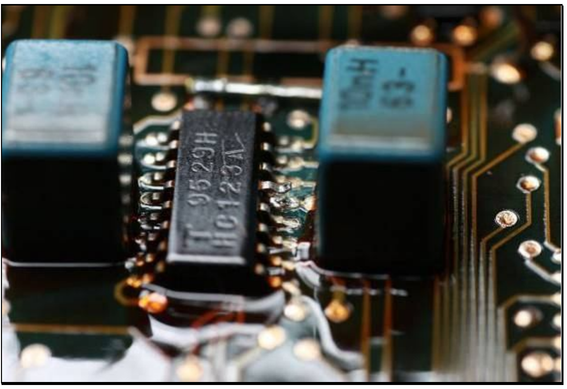
The T5 valve has two coils that control the position of a metal sheet which in turn controls the flow of air through the valve. These two coils are driven by the MOSFET's inside the ECU. The T7 valve on the other hand has only one coil and the control frequency for the T7 valve is slightly different.

Hardware modifications

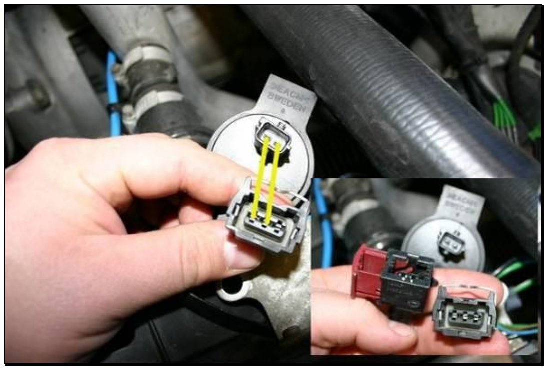
Lift up pin 5 from 74HC123 shown in picture below. Chip is a monostable multivibrator which is used to make two PWM-signals from CPU's original one for that original two coiled valve. Pin 5 is output and pin 12 is inverted output (which we don't need any more). Pin 5 needs to be lifted because otherwise the 74HC14 driving the 74HC123 would be competing with 74HC123's output and either would fan out. This would result in no drive to FET at all. Alternative is to remove the HC123 from the board. The 74HC123 is used in T5 to shorten PWM pulses to 2 ms length by triggering from PWM signal edges. This is enough to make the thin disc inside T5 valve to fly to either end of its movement range. Longer pulses or plain PWM signal could eventually fry a T5 valve. So actually the T5 valve is not driven with PWM at all but with a kind of pulse position modulation. If this is fed to a traditional PWM valve nothing would happen.

Secondly connect the two pads (bridge) between the resistor and the chip with some solder or a small wire. This is connecting pin 5 and pin 11 which is PWM-signal input from CPU. Now the original PWM-signal from CPU is used to drive valve coil. CPU's PWM-signal is wide enough to drive T7 valve. (More resistance > more voltage is needed). This bridge (B01, if you have silk screen printing in your box) was added by Saab in T5.5 to enable bypassing HC123 as a placement option. With this bypass the PWM provided by timer unit in 68332 passes directly to one of the FETs.





Wire the valve as shown in the picture. The 3-pin connector is the original harness of 9k, the 2-pin connector is in the T7 valve.



Add a diode over the valve connector pins as a fail-safe mechanism. Doubt exists about the effectiveness of the internal fly-back diode inside the MOSFETs. This diode will guide reversed polarized voltages past the MOSFET because the MOSFET would "die" if this was not done. These reversed voltages do exists because the load the FET is driving is inductive. This diode must have a reverse recovery time of 75 ns or less and must be able to withstand more than 3 A.

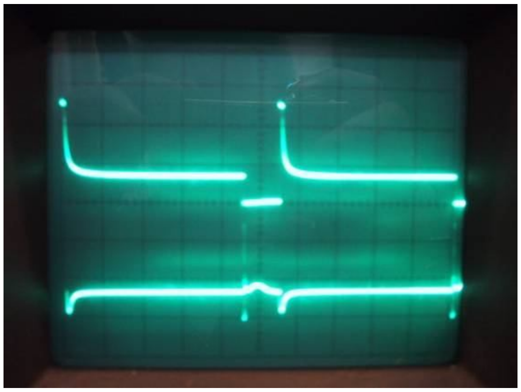
Software modifications

The changes in the software are limited to changing the APC control frequency for the valve. The frequency changes at 2500 rpm to prevent resonation in the vacuum hoses so there's actually to variables to change. Frek_230! holds the control frequency below 2500 rpm. Frek_250! holds the control frequency above 2500 rpm. For the T5 valve these values are set to 90 respectively 70 Hz. For the T7 valve these should be set to 50 and 32 Hz.

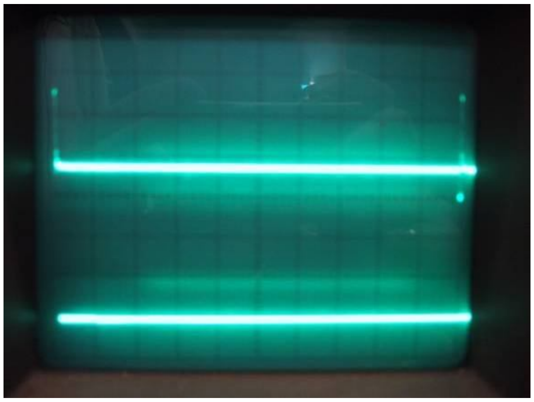
Parameter	Trionic 5 valve	Trionic 7 valve
Frek_230!	90 Hz	50 Hz
Frek_250!	70 Hz	32 Hz

Proving it is possible

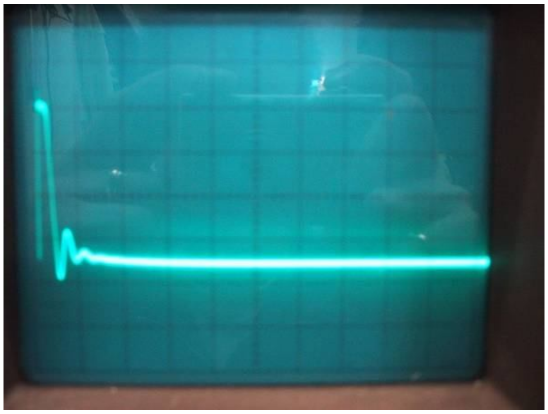
To prove the feasibility of this conversion we need to measure the transient voltages on the valve itself. Using an oscilloscope the transients on both drive wires for the T5 valve are measured and the result is given in the picture. The upper channel is the leftmost connection on the valve while the lower channel is the rightmost connection on the valve. Scaling is 20 volts per division (square) vertically and 2 milliseconds per division horizontally. The car is idling and we can see that the driving frequency is once per (5.5 * 2) 11 ms = 90 Hz which is exactly what we saw earlier in the software.



When the hardware conversion is done and we measure on the T7 pin (there is only one, so second channel is 'dead') we see this result. The upper channel shows the T7 valve coil pull (very short compared to T5 pull). Scaling is the same as the first picture. We see that the frequency has dropped to (10* 2 ms = 20 ms) 50 Hz due to the software modifications we did on the Trionic (90 to 50 Hz below 2500 RPM).



We need to zoom in a little to get a better view on the 'pull spike' that is on the coil.

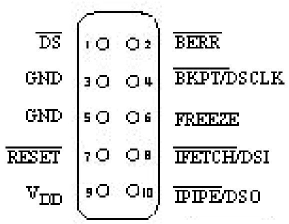


The valve pull is much shorter (this is what we do by the hardware conversion described in the previous paragraph) but the transient response on the coil is practically the same.

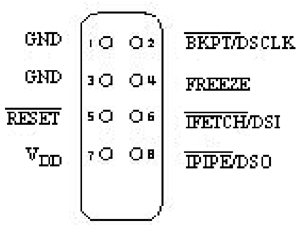
Appendix X: How to connect the PD BDM programmer to a T5/T7 ECU

Pinout

The standard BDM pinout:

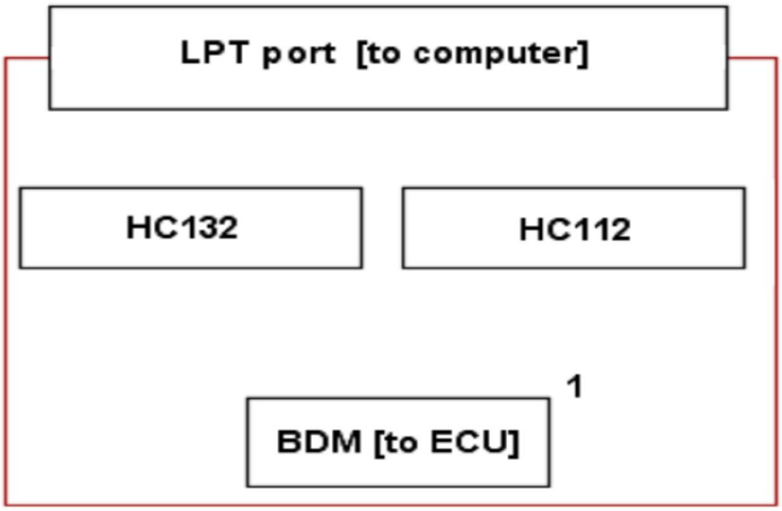


The PD BDM/Willem pinout:



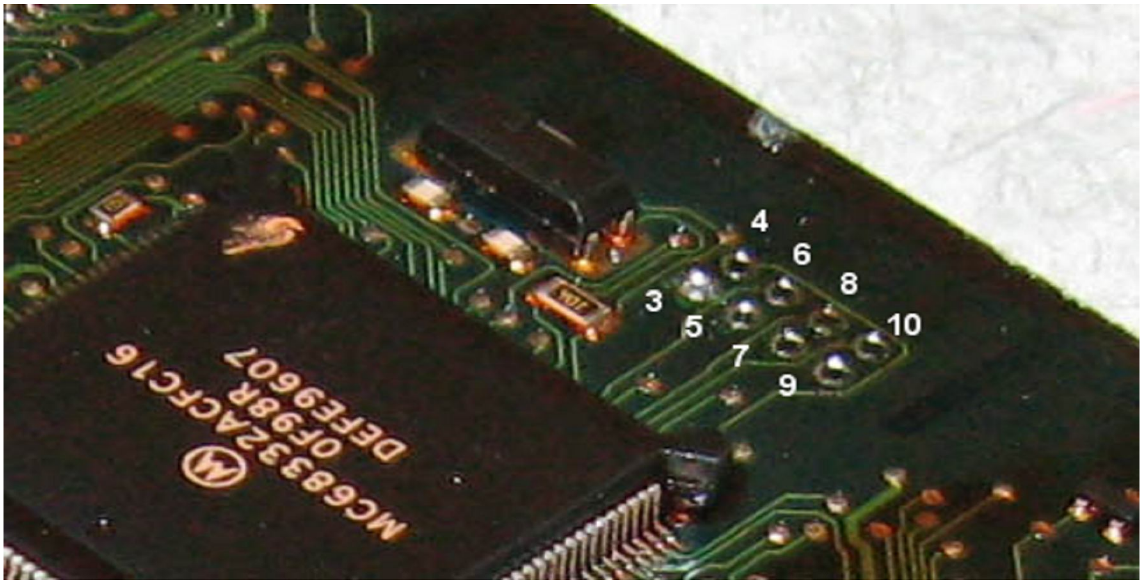
Please note that pins 1 & 2 on a standard BDM connector are not connected and that the row of pins are shifted one pin upwards. Pin 1 on the PD BDM/Willem is pin 3 on the standard BDM connector.

Deltawave BDM layout & pinout



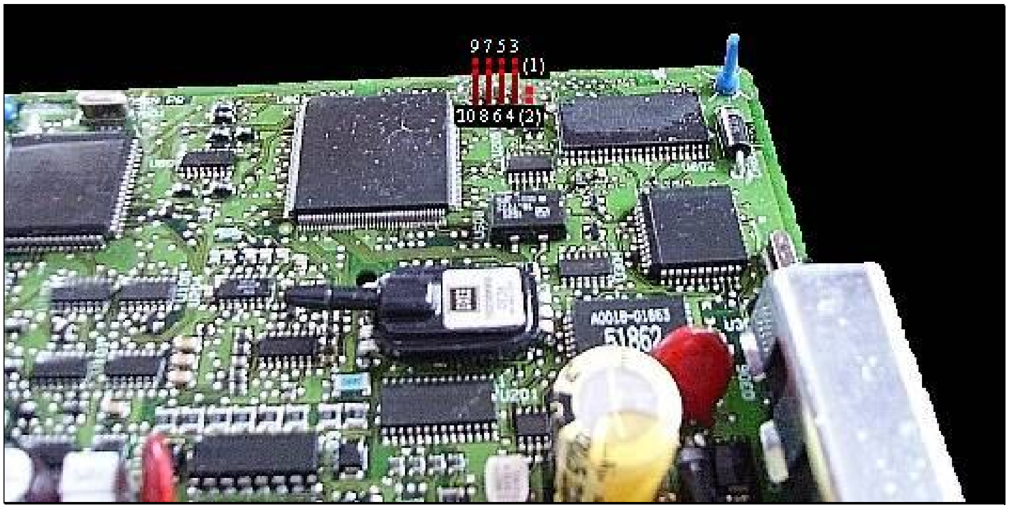
Pin	Description
1	NC – NOT CONNECTED
2	NC – NOT CONNECTED
3	GROUND
4	BKPT
5	NC – NOT CONNECTED
6	FREEZE
7	RESET
8	IFETCH
9	VCC
10	IPIPE

The BDM connector on the T5



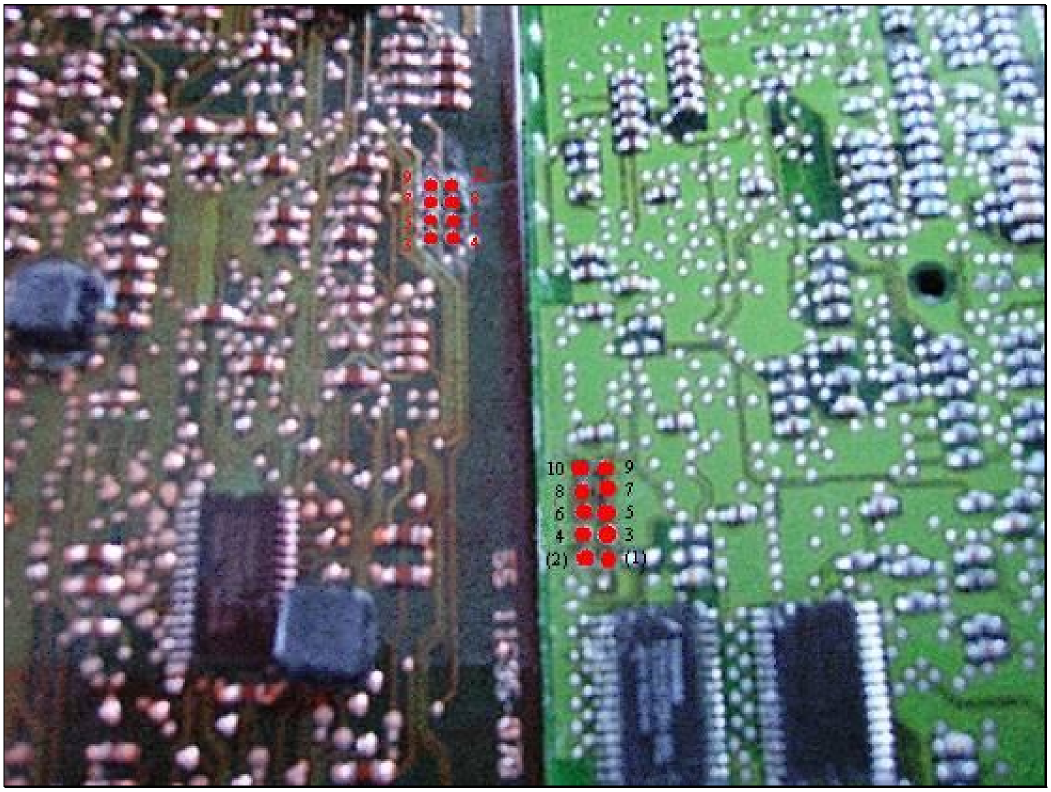
On the T5 ECU there are eight pads for soldering the pins for the BDM programmer.

The BDM connector on the T7



Pay attention to that the BDM connectors are mirrored between the T5/T7 ECU. On the T7 ECU the pads for pin 1 & 2 are present but shall not be used when using a PD BDM or Willem programmer.

The bottom side:



On the other side of the PCB's there are only the other side of the pins. No bridging or something like that, just plain soldering.

Appendix XI: How to flash a T5 ECU

Overview

This how to describes the procedure for writing a binary file to the flash in a T5 ECU. It also describes how to dump the flash memory to a binary file.

Important note:

The flash memories in the T5 ECUs are starting to get old and writing to the flash does not always work. If this should be the case the ECU will be inoperable and the only way to fix it is to physically replace the flashes. Unfortunately there is no way to determine the status of the flash in advance. Please do not use flashT5 if you only have one ECU and you need it to drive your car.

Prerequisites

- BD32 must be installed and working (see Appendix XII).
- The BDM interface must be connected to the T5 ECU and to the computer.
- The T5 ECU must be powered up and +16 volt should be applied to pinnumber 65.

Steps

1. Install FlashT5

- Download FlashT5 from the Downloads section on http://trionic.mobixs.eu
- Unpack the files in the same directory as BD32 is installed in.

2. Prepare ECU for flash management

- In BD32 give the command: "do prepT5.do". This will start a script that prepares the processor in the ECU for flash management. Among several things this script will disable the watchdog, set up registers to allow writing to flash and tell BD32 where to load the target resident program.

3. Dump flash memory

This step is not necessary but can be used to verify that the content of the flash really is changed. Also, it is a good idea to always make a backup of the flash before you start writing to it.

- In BD32 give the command: "dumpT5 filename.bin"

filename.bin should be replaced with a file name of your choice keeping in mind that DOS can only handle 8.3 filename format (8 characters for filename and 3 for extension). BD32 will now download a target resident program that reads from flash and sends the information to BD32. You will see a number of dots appearing in BD32 during the copying.

4. Flash the ECU

- Read Important note above!
- In BD32 give the command: "flashT5 filename.bin"

filename.bin should be replaced by the name of the binary file that you want to write to flash keeping in mind that DOS can only handle 8.3 filename format (8 characters for filename and 3 for extension). This file should be in the same folder as the BD32 software. BD32 will now download a target resident program called flashT5.d32 to the processor in the ECU and execute it. The target resident program and BD32 cooperates during the flashing (BD32 takes care of file handling and the target resident program handles writing to flash). During the flashing you will see a number of dots being printed in BD32. This will stop when the flashing is completed and the command prompt of BD32 will appear.

5. If you run into a problem

If you run into problem when you are flashing try this:

- Double the delay parameter.
- The bottom field in BD32 should report "MCU: STOPPED" while you are using flashT5. If you for some reason have given the reset command you need to run the prepT5.do script again in order to setup the registers and stop the MCU.
- If you get an Erase Error or Program Error message it is likely that your flash cannot be written to (see Important note above). You can try flashing a couple of more times but it is likely that the ECU can no longer be used.

Appendix XII: Howto install BD32 with a PD BDM interface

Overview

This Howto describes how to install BD32 (found at http://trionic.mobixs.eu) with a PD (public domain) BDM interface. The software will also work with an ICD interface but this is not covered in this Howto.

Steps

1. Install BD32

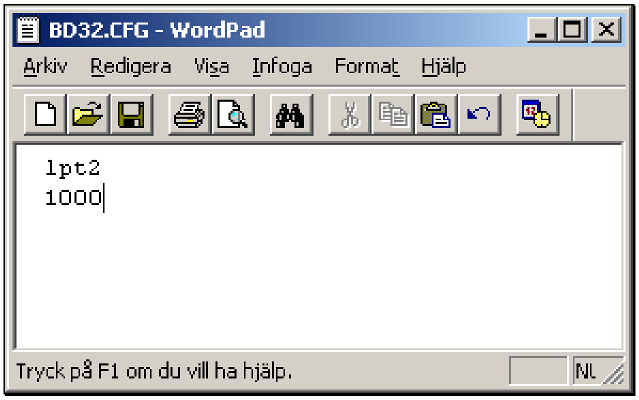
Download bd32-122.zip from http://trionic.mobixs.eu and unzip it on your hard drive.

2. Connect BDM interface

Connect the BDM interface to the parallel port and to the target device (for example your T5 ECU). You need to know which port you have connected the interface to (LPT1 or LPT2).

3. Edit BD32.CFG

The BD32.CFG contains settings for BD32.EXE. The first line tells which parallel port the BDM interface is connected to. The second line sets a delay parameter that adjust the speed of BD32 to the interface (depends on how fast the computer is). This parameter will be tuned later in this Howto. For now, set it to 1000. Your BD32.CFG should now look something like this:


BD32.CFG - WordPad
Arkiv Redigera Visa Infoga Format Hjälp
Tryck på F1 om du vill ha hjälp.
NL

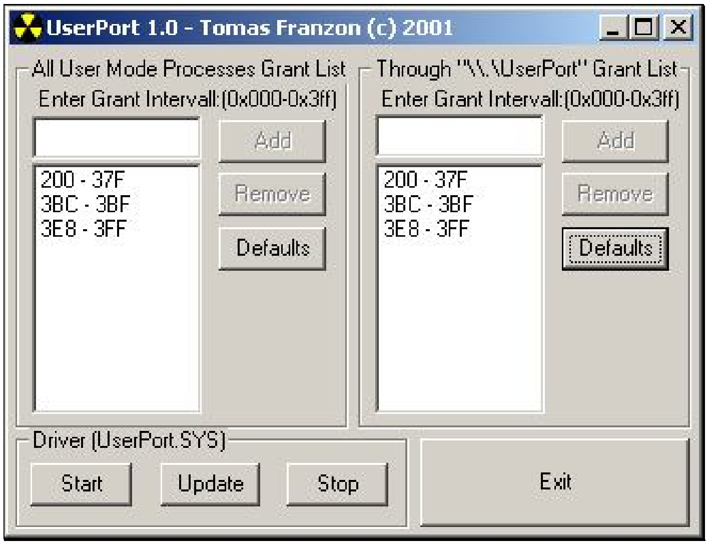
```
lpt2
1000
```

In this case the BDM interface is connected to LPT2.

4. Install UserPort

This step is only needed if you are using Windows 2000 or later.

- Download UserPort.zip and unzip it on your hard drive.
- Copy the file UserPort.sys to %windows-base%\system32\drivers where %windows-base% is the installation folder of windows (typically c:\windows). Example c:\windows\system32\drivers.
- Start UserPort.exe. You will see something like this:


UserPort 1.0 - Tomas Franzon (c) 2001
All User Mode Processes Grant List
Enter Grant Intervall:(0x000-0x3ff)
Add
Remove
Defaults
Through "\\.\UserPort" Grant List
Enter Grant Intervall:(0x000-0x3ff)
Add
Remove
Defaults
Driver (UserPort.SYS)
Start
Update
Stop
Exit

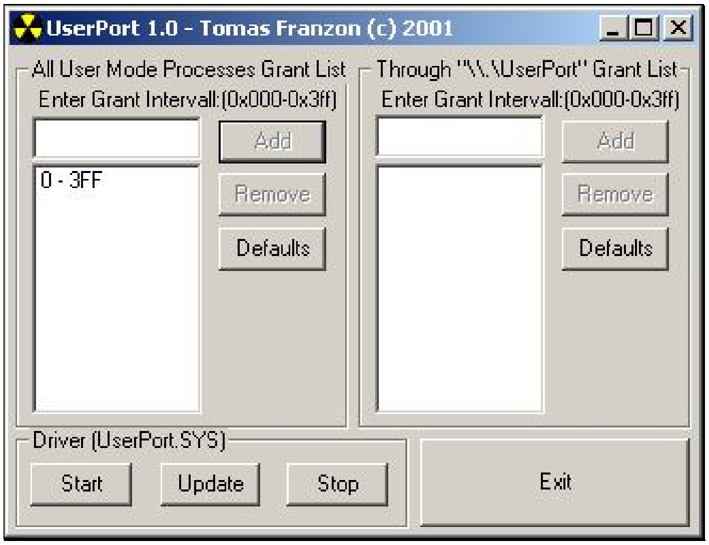
All User Mode Processes Grant List:

200 - 37F
3BC - 3BF
3E8 - 3FF

Through "\\.\UserPort" Grant List:

200 - 37F
3BC - 3BF
3E8 - 3FF

Here you can limit the address range to open up if you know what address your parallel port hardware has. If you don't know or don't care you can open up the whole range from 0x000 to 0x3ff like this:


UserPort 1.0 - Tomas Franzon (c) 2001
All User Mode Processes Grant List
Enter Grant Intervall:(0x000-0x3ff)
Add
Remove
Defaults
Through "\\.\UserPort" Grant List
Enter Grant Intervall:(0x000-0x3ff)
Add
Remove
Defaults
Driver (UserPort.SYS)
Start
Update
Stop
Exit

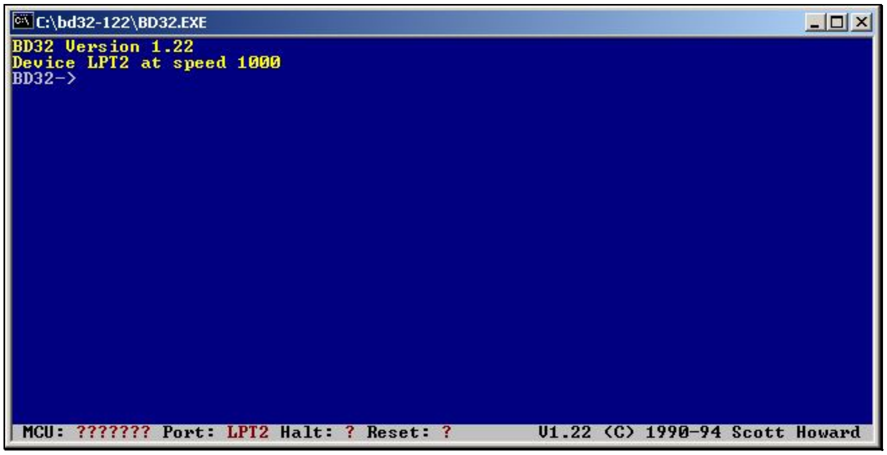
All User Mode Processes Grant List:

0 - 3FF

- Now click on Start to activate UserPort.

5. Start BD32

- If you haven't already powered up your target device do it now. Then start BD32.EXE. If everything works you will see something like this:



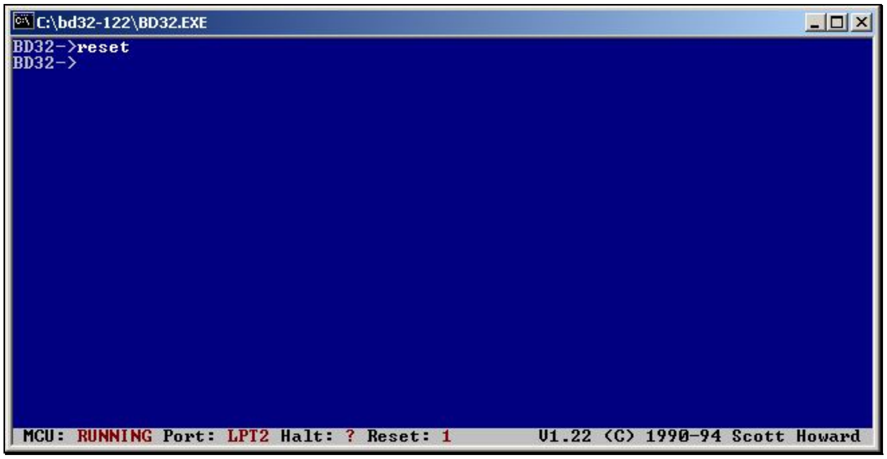
```
BD32 Version 1.22
Device LPT2 at speed 1000
BD32->
```

```
MCU: ??????? Port: LPT2 Halt: ? Reset: ?          V1.22 (C) 1990-94 Scott Howard
```

6. Set the delay parameter

Now it is time to adjust the timing between the BDM interface and your computer (this is not needed if you use an ICD interface).

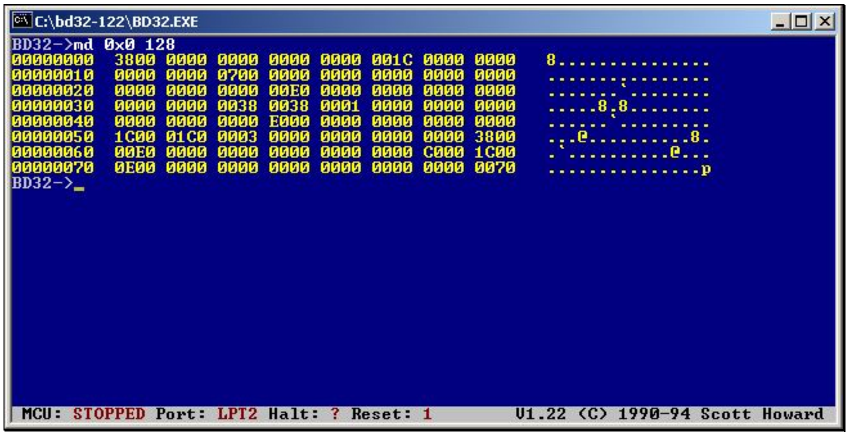
- First give the reset command:



```
BD32->reset
BD32->
```

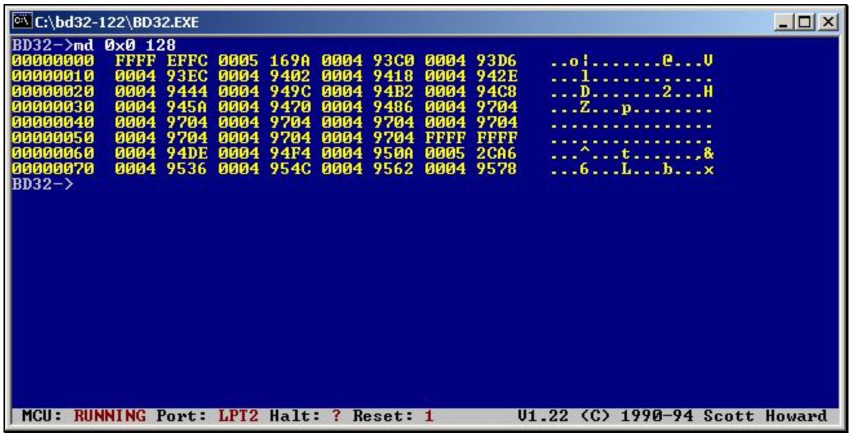
```
MCU: RUNNING Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```

- In BD32 give the command: "md 0x0 128". This will print the first 128 bytes of the flash memory. If the timing is not correct it might look something like this:



```
C:\bd32-122\BD32.EXE
BD32->md 0x0 128
00000000  3800 0000 0000 0000 0000 001C 0000 0000    8...............
00000010  0000 0000 0700 0000 0000 0000 0000 0000    ................
00000020  0000 0000 0000 00E0 0000 0000 0000 0000    .......`........
00000030  0000 0000 0038 0038 0001 0000 0000 0000    .....8.8........
00000040  0000 0000 0000 E000 0000 0000 0000 0000    ......`.........
00000050  1C00 01C0 0003 0000 0000 0000 0000 3800    ...@..........8.
00000060  00E0 0000 0000 0000 0000 0000 C000 1C00    .`..........@...
00000070  0E00 0000 0000 0000 0000 0000 0000 0070    ...............p
BD32->_

 MCU: STOPPED Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```

But it should look like this (if you connect to a T7 ECU):

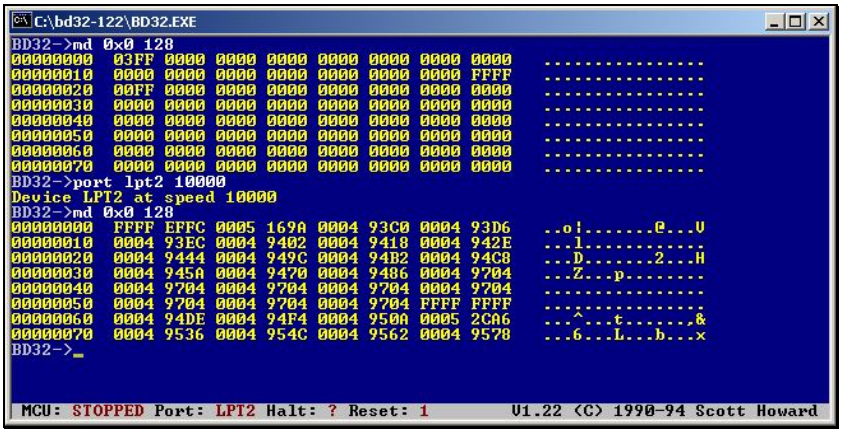


```
C:\bd32-122\BD32.EXE
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6    ..o|.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E    ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8    ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704    ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704    ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF    ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6    ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578    ...6...L...b...x
BD32->

 MCU: RUNNING Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```

A T7 ECU always starts with "FFFF EFFC" while a T5 always starts with “FFFF F7FC”. Most important is that each time you give the command “reset” followed by "md 0x0 128" the result should be the same.

Now, if what you see does not start with "FFFF EFFC" or “FFFF F7FC” or if you get different result each time you give the command "md 0x0 128" you need to increase the delay parameter. The initial value we set to 1000.

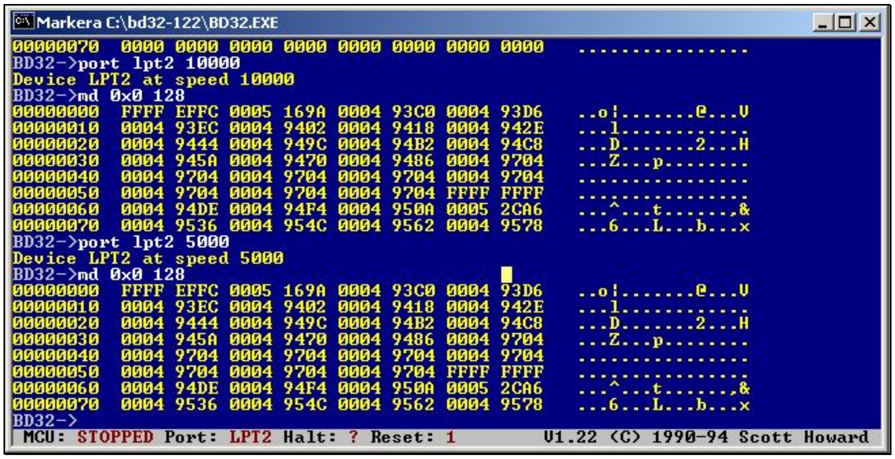
- Start by adding a zero to that number by giving the command: "port lptN 10000" (where N is replaced with the port number). Then give the command "md 0x0 128". For example:



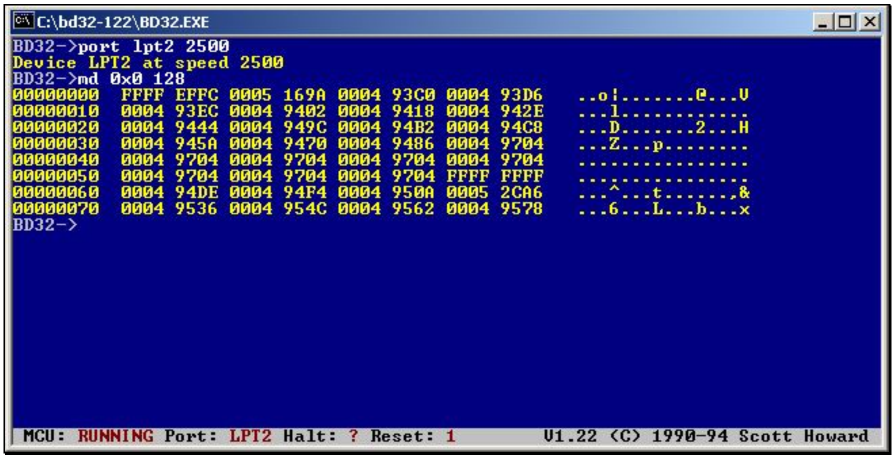
```
C:\bd32-122\BD32.EXE
BD32->md 0x0 128
00000000  03FF 0000 0000 0000 0000 0000 0000 0000    ................
00000010  0000 0000 0000 0000 0000 0000 0000 FFFF    ................
00000020  00FF 0000 0000 0000 0000 0000 0000 0000    ................
00000030  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000040  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000050  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000060  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000070  0000 0000 0000 0000 0000 0000 0000 0000    ................
BD32->port lpt2 10000
Device LPT2 at speed 10000
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6    ..o|.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E    ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8    ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704    ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704    ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF    ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6    ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578    ...6...L...b...x
BD32->_

 MCU: STOPPED Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```

In this case a delay of 10000 seems to be enough but if you have a fast computer you may need to increase this number.

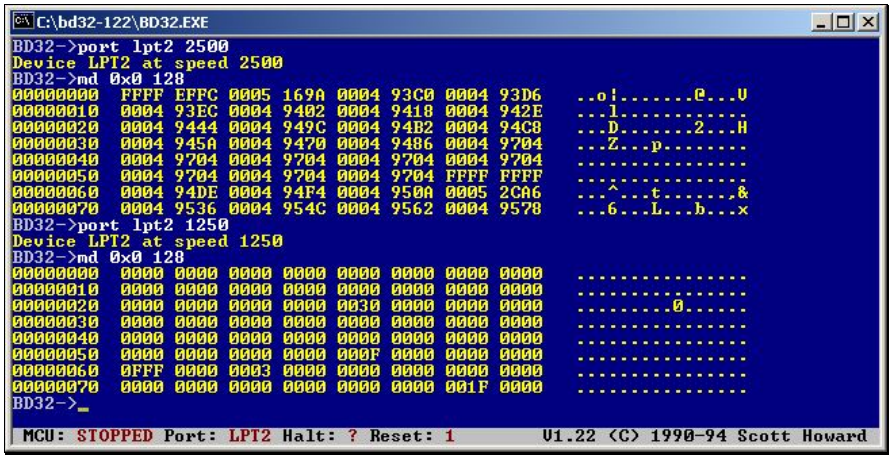
Now it is time to fine tune the delay parameter. Halving the delay parameter and give the command "reset" and "md 0x0 128". Repeat this until you get an uncertain result and then increase it again. Example:



```
Markera C:\bd32-122\BD32.EXE
00000070  0000 0000 0000 0000 0000 0000 0000 0000    ................
BD32->port lpt2 10000
Device LPT2 at speed 10000
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6    ..o¦.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E    ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8    ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704    ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704    ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF    ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6    ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578    ...6...L...b...x
BD32->port lpt2 5000
Device LPT2 at speed 5000
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6    ..o¦.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E    ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8    ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704    ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704    ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF    ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6    ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578    ...6...L...b...x
BD32->
 MCU: STOPPED Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```



```
C:\bd32-122\BD32.EXE
BD32->port lpt2 2500
Device LPT2 at speed 2500
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6    ..o¦.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E    ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8    ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704    ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704    ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF    ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6    ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578    ...6...L...b...x
BD32->
 MCU: RUNNING Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```



```
C:\bd32-122\BD32.EXE
BD32->port lpt2 2500
Device LPT2 at speed 2500
BD32->md 0x0 128
00000000  FFFF EFFC 0005 169A 0004 93C0 0004 93D6    ..o¦.......@...V
00000010  0004 93EC 0004 9402 0004 9418 0004 942E    ...l............
00000020  0004 9444 0004 949C 0004 94B2 0004 94C8    ...D.......2...H
00000030  0004 945A 0004 9470 0004 9486 0004 9704    ...Z...p........
00000040  0004 9704 0004 9704 0004 9704 0004 9704    ................
00000050  0004 9704 0004 9704 0004 9704 FFFF FFFF    ................
00000060  0004 94DE 0004 94F4 0004 950A 0005 2CA6    ...^...t......,&
00000070  0004 9536 0004 954C 0004 9562 0004 9578    ...6...L...b...x
BD32->port lpt2 1250
Device LPT2 at speed 1250
BD32->md 0x0 128
00000000  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000010  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000020  0000 0000 0000 0000 0030 0000 0000 0000    .........0......
00000030  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000040  0000 0000 0000 0000 0000 0000 0000 0000    ................
00000050  0000 0000 0000 0000 000F 0000 0000 0000    ................
00000060  0FFF 0000 0003 0000 0000 0000 0000 0000    ................
00000070  0000 0000 0000 0000 0000 0000 001F 0000    ................
BD32->_
 MCU: STOPPED Port: LPT2 Halt: ? Reset: 1          V1.22 (C) 1990-94 Scott Howard
```

Decreasing the delay parameter to 1250 in this example seems to give a very uncertain result so therefore we set it to 2500. It is possible to fine tune it even further but don't try to set it close to the uncertainty level. Better be safe.

Typical delay values for some computers:

Pentium I 90 MHz: 200
Pentium III 800 MHz: 2500

7. Update BD32.CFG

Update the delay parameter in BD32.CFG with the value that you found in 6. This will make sure that you get the correct setting every time you start BD32.

8. Start using BD32

Now everything is set up for using BD32. Read the document BD32.DOC if you want to know more about the debugger or start using the flash tools that can be found in the Downloads section of http://trionic.mobixs.eu

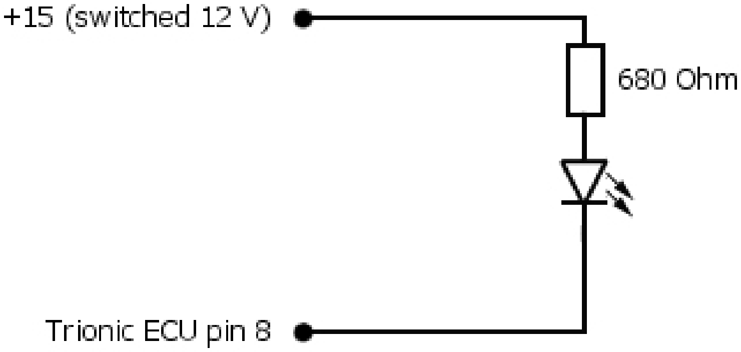
Appendix XIII: Installing a knock LED

Description

This chapter will describe the installation procedure for a knock LED. This LED will indicate detected knock to the driver. Trionic has a "service" output that switches to ground in the event of knocking detection. This signal is available on pin 8 on the ECU 70-pin connector. Since the output is normally 'floating' and switches to ground when the knock detection occurs we can connect the ground side (cathode) of a LED to the ECU's output and connect the LED's "hot" side (anode) via a resistor (680 Ω - 1 kΩ) to +12V (switched over ignition).

Schematic diagram

The image below shows a schematic diagram for this.



A normal LED (light emitting diode) has a long lead and a short lead. The long lead is the anode which is the positive side (current flows into the diode here) while the short lead is the cathode which is the negative side (current flows out of the diode here). The top side of the diode in the schematic diagram is the anode and the bottom side is the cathode. Note: On 5mm LEDs the cathode side is also indicated by the housing being flat on the side of this lead.

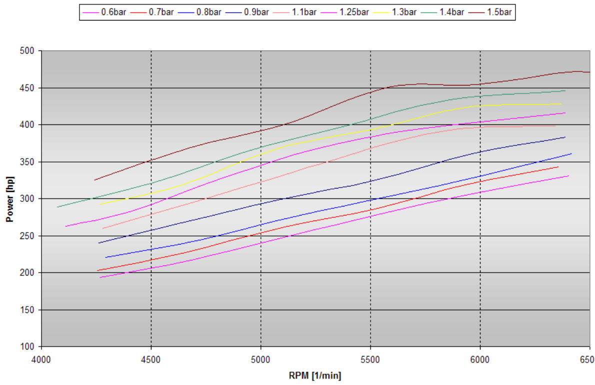
Do not attempt to hook up a normal lamp to the ECU's output because the ECU cannot sink so much current from it.

Appendix XIV: A tuned example on the road

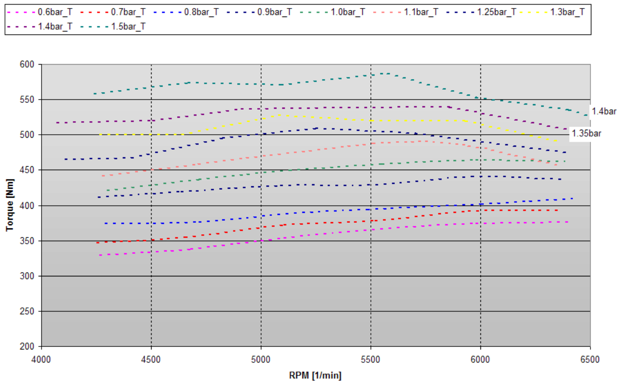
Description

In this chapter we will see some measured data from a tuned car which will give more insight in how boostlevels relate to power, torque and acceleration. Please do note that these figures are just give to illustrate and that the results of tuning will vary for every engine and car.

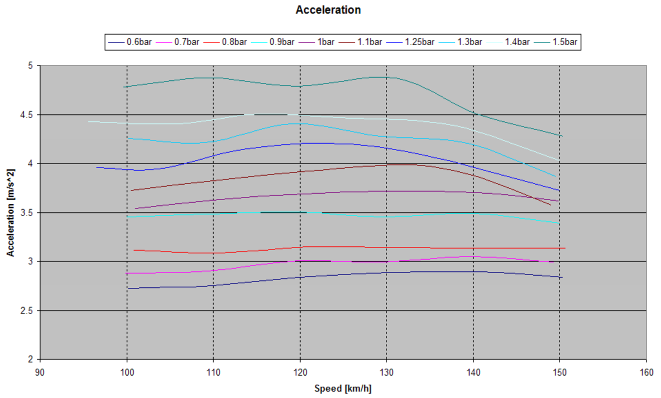
Here we see the power curves in brake horsepower related to different boostvalues…



and the torque curves in Newton meters related to boost…



And last but not lease the acceleration values from 100-150 km/h related to the same boost levels...



Appendix XV: Program mode status bytes

Description

Byte	Bitmask	Description
0	0x01	After start enrichment
0	0x02	Wide open throttle enrichment
0	0x04	Interpolation of delay
0	0x08	Temperature compensation
0	0x10	Lambda control
0	0x20	Adaptivity (adaption enabled)
0	0x40	Idle control
0	0x80	Enrichment during start
1	0x01	Constant injection time E51
1	0x02	Lambda control during transients
1	0x04	Fuelcut
1	0x08	Constant injection time during idle
1	0x10	Acceleration enrichment
1	0x20	Deceleration enleanment
1	0x40	
1	0x80	Adaptivity with closed throttle
2	0x01	Factor to lambda control when throttle is opening
2	0x02	Use separate injection map during idle
2	0x04	Factor to lambda control when AC is engaged
2	0x08	Throttle Acc ret adjust simultaneously MY1995
2	0x10	Fuel adjust during idle
2	0x20	Purge enable
2	0x40	Adaption of idle control
2	0x80	Lambda control during idle
3	0x01	Heat plates compensation
3	0x02	Automatic transmission
3	0x04	Load control
3	0x08	ETS
3	0x10	APC control
3	0x20	Higher idle during start
3	0x40	Global adaption enable
3	0x80	Temperature compensation with active lambda control
4	0x01	Load buffer during idle
4	0x02	Constant idle ignition angle in gears one and two
4	0x04	No fuel cut in reverse, first and second gear
4	0x08	Airpump control
4	0x10	Normally aspirated engine
4	0x20	Knock regulation disabled
4	0x40	Constant ignition angle
4	0x80	Purge valve MY1994
5	0x01	
5	0x02	
5	0x04	
5	0x08	
5	0x10	
5	0x20	
5	0x40	
5	0x80	

Appendix XVI: Intercooler calculation

Description

This appendix will explain how to do calculations on intercooler flow capacity. A larger than stock intercooler is needed of you plan to go over 300 bhp with a Trionic 5 engine.

An intercooler is a heat exchanger. That means there are two or more fluids or gases that don't physically touch each other but a transfer heat or energy takes place between them. At wide open throttle and full boost the hot compressed air coming from a turbocharger is probably between 250 and 350 °F depending on the particular turbo, boost pressure, outside air temperature, etc.. We want to cool it down, which reduces its volume so we can pack more air molecules into the cylinders and reduce the engine's likelihood of detonation.

How does an intercooler work? Hot air from the turbo flows through tubes inside the intercooler. The turbo air transfers heat to the tubes, warming the tubes and cooling the turbo air. Outside air (or water in a watercooler intercooler) passes over the tubes and between fins that are attached to the tubes. Heat is transferred from the hot tubes and fins to the cool outside air. This heats the outside air while cooling the tubes. This is how the turbo air is cooled down. Heat goes from the turbo air to the tubes to the outside air.

There are some useful equations which will help us understand the factors involved in transferring heat. After we look at these equations and see what's important and what's not, we can talk about what all this means.

Equation 1

The first equation describes the overall heat transfer that occurs.
Q = U x A x DTlm

Q is the amount of energy that is transferred.
U is called the heat transfer coefficient. It is a measure of how well the exchanger transfers heat. The bigger the number, the better the transfer.
A is the heat transfer area, or the surface area of the intercooler tubes and fins that is exposed to the outside air.
DTlm is called the log mean temperature difference. It is an indication of the "driving force", or the overall average difference in temperature between the hot and cold fluids. The equation for this is:

$$\mathrm{DTlm} = \frac{(\mathrm{DT1}-\mathrm{DT2})\ \ast\ F}{\ln(\mathrm{DT1}/\mathrm{DT2})}$$

where **DT1** = turbo air temperature in - outside air temperature out
DT2 = turbo air temperature out - outside air temperature in
F = a correction factor, see below

Note:

The outside air that passes through the fins on the passenger side of the intercooler comes out hotter than the air passing through the fins on the driver’s side of the intercooler. If you captured the air passing through all the fins and mixed it up, the temperature of this mix is the "outside air temperature out".

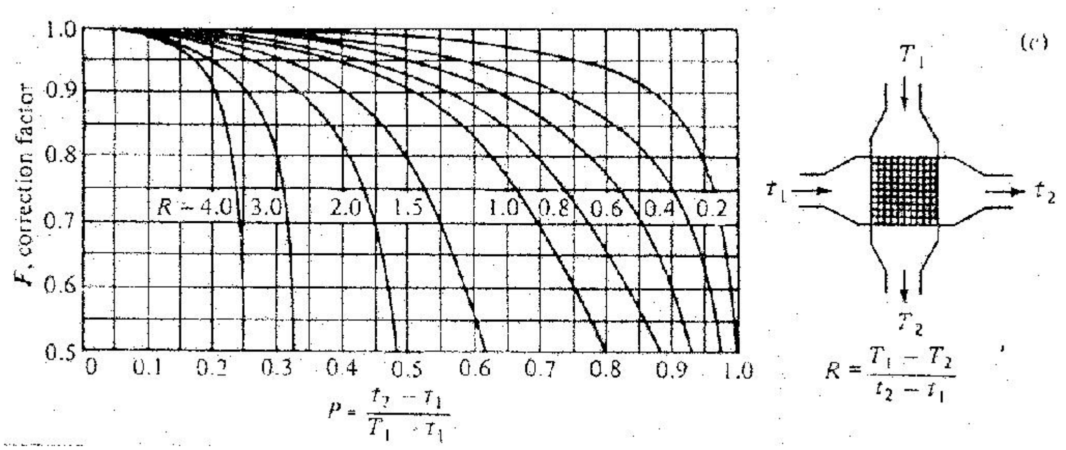
F is a correction factor that accounts for the fact that the cooling air coming out of the back of the intercooler is cooler on one side than the other.

To calculate this correction factor, calculate "P" and "R":

$$P = \frac{\text{turbo air temp out - turbo air temp in}}{\text{outside air temp in - turbo air temp in}}$$

$$R = \frac{\text{outside air temp in - outside air temp out}}{\text{turbo air temp out - turbo air temp in}}$$

Find P and R on "Fchart.jpg" (below) and read F off the left hand side.



This overall heat transfer equation shows us how to get better intercooler performance. To get colder air out of the intercooler we need to transfer more heat, or make Q bigger in other words. To make Q bigger we have to make U, A, or DTlm bigger, so that when you multiply them all together you get a bigger number. More on that later.

Equation 2

We also have an equation for checking the amount of heat lost or gained by the gas (or fluid) on one side of the heat exchanger (i.e., just the turbo air or just the outside air):

$$Q = m \times Cp \times DT$$

Q is the energy transferred. It will have the exact same value as the Q in the first equation. If 5000 BTU are transferred from turbo air to outside air, then Q = 5000 for this equation AND the first equation.
m is the mass flowrate (lbs/minute) of fluid, in this case either turbo air or outside air depending on which side you're looking at.
Cp is the heat capacity of the air. This is a measure of the amount of energy that the fluid will absorb for every degree of temperature that it goes up. It is about 0.25 for air and 1.0 for water. Air doesn't do a great job of absorbing heat. If you put 10 BTU into a pound of air the temperature of it goes up about 40 degrees. If you put 10 BTU into a pound of water, the temperature only goes up about 10 degrees! Water is a great energy absorber. That's why we use water for radiators instead of some other fluid.
DT is the difference in temperature between the inlet and outlet. If the air is 200 deg going in and 125 deg coming out, then DT = 200 - 125 = 75. Again, on the cooling air side the outlet temperature is the average "mix" temperature.

If you know 3 of the 4 main variables on one side of the exchanger (the amount of heat transferred, the inlet and outlet temperatures, and the fluid's flow rate) then this equation is used to figure out the 4th. For example, if you know the amount of heat transferred, the inlet temperature, and the flow rate you can calculate the outlet temperature. Since you can't measure everything, this equation is used to figure out what you don't know.

Caveat:
These equations are all for steady state heat transfer, which we probably don't really see too much under the conditions that we are most interested in – turbocharged engines! Cruising on the highway you would definitely see steady state.

So, now that we've got these equations, what do they **REALLY** tell us?

1. Heat transfer goes really well when there is a large temperature difference, or driving force, between the two gases. This is shown in equation 1 as a large DTlm. It doesn't go as well when there is a small temperature difference between the two gases (small DTlm). The closer you get the intercooler outlet temperature to the outside air temperature the smaller DTlm gets, which makes the heat transfer tougher.

2. The difference between the intercooler outlet temperature and the outside air temperature is called the approach. If it is 100 degrees outside and your intercooler cools the air going into the intake manifold down to 140 degrees, then you have an approach of 40 degrees (140 - 100 = 40). To get a better (smaller) approach you have to have more area or a better U, but there is a problem with diminishing returns. Lets rearrange the first equation to Q/DTlm = U x A. Every time DTlm goes down (get a better temperature approach) then Q goes up (transfer more heat, get a colder outlet temperature), and dividing Q by DTlm gets bigger a lot faster than U x A does. The upshot of that is we have a situation of diminishing returns; for every degree of a better approach you need more and more U x A to get there. Start with a 30 deg approach and go to 20 and you have to improve U x A by some amount, to go from 20 to 10 you need to increase U x A by an even bigger amount.

3. I would consider an approach of 20 degrees to be pretty good. In industrial heat exchangers it starts to get uneconomical to do better somewhere around there, the exchanger starts to get too big to justify the added expense. The one time I checked my car (stock turbo, stock IC, ported heads, bigger cam) I had an approach of about 60 deg. The only practical way of making the DTlm bigger on an existing intercooler is to only drive on cold days; if you buy a better intercooler you naturally get a better DTlm.

4. You can transfer more heat (and have cooler outlet temps) with more heat transfer area. That means buying a new intercooler with more tubes, more fins, longer tubes, or all three. This is what most aftermarket intercoolers strive for. Big front mounts, intercooler and a half, etc... are all increasing the area.

   A practical consideration is the fin count. The area of the fins is included in the heat transfer area; more fins means more area. If you try to pack too many fins into the intercooler the heat transfer area does go up, which is good, but the cooling air flow over the fins goes down, which is bad. Looking at the 2nd equation, Q = m * Cp * DT, when the fin count is too high then the air flow ("m") drops. For a given Q that you are trying to reach then you have to have a bigger DT, which means you have to heat up that air more. Then THAT affects the DTlm in the first equation, making it smaller, and lowering the overall heat transfer. So there is an optimum to be found. Starting off with bare tubes you add fins and the heat transfer goes up because you're increasing the area, and you keep adding fins until the it starts to choke off the cooling air flow and heat transfer starts going back down. At that point you have to add more tubes or make them longer to get more heat transfer out of the increased area.

5. Make U go up. You can increase the U by adding or improving "turbulators" inside the tubes. These are fins inside the tubes which cause the air to swirl inside the tube and makes it transfer its heat to the tube more efficiently. Our intercoolers have these, but I understand that more efficient designs are now available. One of the best ways to increase the U is to clean the tubes out! Oil film (from a bad turbo seal or from the stock valve cover breather) inside the tubes acts as an insulator or thermal barrier. It keeps heat from moving from the air to the tube wall. This is expressed in our equation as a lower U. Lower U means lower Qs which mean hotter turbo air temperatures coming out of the intercooler.

6. Air-to-water. If we use water as the cooling medium instead of outside air, we can see a big improvement for several reasons: Water can absorb more energy with a lower temperature rise. This improves our DTlm, makes it bigger, which makes Q go up and outlet temps go down. A well designed water cooled exchanger also has a much bigger U, which also helps Q go up. And since both DTlm and U went up, you can make the area A smaller which makes it easier to fit the intercooler in the engine compartment. Of course, there are some practical drawbacks. The need for a water circulation system is one. A big one is cooling the water down after it is heated (which means another radiator). This leads to another problem: You heat the water, and cool it down with outside air like the Syclone/Typhoon. You can't get it as cool as the outside air, but maybe you can get it within 20 degrees of it. Now you are cooling the turbo air with water that is 20 hotter than the outside air, and you can only get within 15 degrees of that temperature so coming out of the intercooler you have turbo air that is 35 degrees hotter than outside! (turbo air is 15 deg over water temp which is 20 deg over outside temp). You could have easily done that with an air to air intercooler! But... if you put ice water in your holding tank and circulate that... Then maybe the air temp coming out of the intercooler is 15 deg above that or 45 to 50 deg. Hang on! But after the water warms up, you're back to the hot air again. So, great for racing, not as good for the street.

7. Lower the inlet temperature. The less hard the turbo has to work to compress the air then the lower the temperature the air coming out of the turbo is. This actually hurts the DTlm, but still if it's cooler going in it will be cooler coming out. You can work the turbo less hard by running less boost, by improving the pressure drop between the air filter and the turbo, or by having a more efficient compressor wheel. You can also reduce the pressure drop in the intercooler, which allows you to run the same amount of boost in the intake manifold while having a lower turbo discharge pressure. More on this later. If you can drop the turbo outlet pressure by 2 psi, or raise the turbo inlet pressure by 1 psi, that will drop the turbo discharge temperature about 16 degrees (depending on the compression efficiency and boost level). If the turbo air is going into the intercooler 16 degrees colder then it may come out only 10 degrees colder than before, but that is still better than what it was.

Pressure drop

Another aspect of intercoolers to be considered is pressure drop. The pressure read by a boost gauge is the pressure in the intake manifold. It is not the same as the pressure that the turbocharger itself puts out. To get a fluid, such as air, to flow there must be a difference in pressure from one end to the other. Consider a straw that is sitting on the table. It doesn't having anything moving through it until you pick it up, stick it in your mouth, and change the pressure at one end (either by blowing or sucking). In the same way the turbo outlet pressure is higher than the intake manifold pressure, and will always be higher than the intake pressure, because there must be a pressure difference for the air to move.

The difference in pressure required for a given amount of air to move from turbo to intake manifold is an indication of the hydraulic restriction of the intercooler, the up pipe, and the throttle body. Let's say you are trying to move 255 gram/sec of air through a stock intercooler, up pipe, and throttle body and there is a 4 psi difference that is pushing it along. If your boost gauge reads 15 psi, that means the turbo is actually putting up 19 psi. Now we increase the amount of air travelling though to 450 gram/sec of air. At 15 psi boost in the intake manifold the turbo now has to put up 23 psi, because the pressure drop required to get the higher air flow is now 8 psi instead of the 4 that we had before. More flow with the same equipment means higher pressure drop. So we put on a new front mount intercooler. It has a lower pressure drop, pressure drop is now 4 psi, so the turbo is putting up 19 psi again. Now we add the larger throttle body and the pressure drop is now 3 psi. Then we add the 3" up pipe, and it drops to 2.5 psi. Now to make 15 psi boost the turbo only has to put up 17.5 psi. The difference in turbo outlet temperature between 23 psi and 17.5 psi is about 40 deg (assuming a constant efficiency)! So you can see how just by reducing the pressure drop we can lower the temperatures while still running the same amount of boost.

Pressure drop is important because the higher the turbo charging pressure is, the higher the temperature of the turbo air. When we drop the turbo charging pressure we also drop the temperature of the air coming out of the turbo. When we do that we also drop the intercooler outlet temperature, although not as much, but every little bit helps. This lower pressure drop is part of the benefit offered by new, bigger front mount intercoolers; by bigger up pipes; and by bigger throttle bodies. You can also make the turbo work less hard by improving the inlet side to it. K&N air filters, these all reduce the pressure drop in the turbo inlet system which makes the compressor work less to produce the same boost which will reduce the turbo charge temperature.

Appendix XVII: Fuel calculation formulas

Correction factors

Here are the formulas that Trionic's injection calculation uses to achieve the final correction coefficient that shapes the fuel correction in each step of the calculation:

Factor	Formula	Range	Weight
Global fuel adaptation	(Adapt_injfaktor+384)/512		
Idle fuel correction	(Idle_fuel_korr+128)/256		
Idle fuel adaptation	(Adapt_inj_imat+384)/512		
Main fuel correction	(Insp_mat+128)/256		
Spot fuel adaptation	(Adapt_korr+128)/256		
Knock fuel correction	(Fuel_knock_mat+128)/256		
Full load correction	(fload_tab+512)/512		
Cylinder compensation correction	(Cyl_komp[Cyl_nr]+384)/512		
Lambdaint Correction	(Lambdaint+384)/512		
Load/TPS acceleration/deceleration correction	(Lacc_konst+256)/256 (Accell_konst+256)/256 (Lret_konst+256)/256 (retard_konst+256)/256 ((Ret_fuel_fak+1024)*fuel_tmp_cor[cyl_nr])/1024 ((Hot_start_fak+1024)*fuel_tmp_cor[cyl_nr])/1024 ((After_fcut+1024)*fuel_tmp_cor[cyl_nr])/1024		

The corresponding map value is always interpolated from the table and then this scaling calculation is done to them giving each step a certain weight. For example acceleration enrichment is never under 1,00 which is logical.
Trionic starts off by calculating base injection time (based on IAT and MAP and Inj_konst) and then in a sequence goes through different corrections depending on the driving conditions etc. For example the Insp_mat (it's not VE but of course depends on the physical VE characteristics of the engine) and Knock_fuel_mat are examples of these. These corrections multiply the injection time (fuel_tmp_cor[cyl_nr]) in each step, so if you have 128 in your Insp_mat, you end up multiplying with 1,00.
Simplified example Insp_mat + Lacc_konst, base time = a, final fuel = x:
x = a
x = ((Insp_mat+128)/256) * x
x = ((Lacc_konst+256)/256) * x

Appendix XVIII: You can be too rich

Adding fuel too cool

By Klaus Allmendinger, VP of Engineering, Innovate Motorsports

Many people with turbochargers believe that they need to run at very rich mixtures. The theory is that the excess fuel cools the intake charge and therefore reduces the probability of knock. It does work in reducing knock, but not because of charge cooling. The following little article shows why.

First let's look at the science. Specific heat is the amount of energy required to raise 1 kg of material by one degree K (Kelvin, same as Celsius but with 0 point at absolute zero). Different materials have different specific heats. The energy is measured in kJ or kilojoules:

Air ~ 1 kJ/(kg * deg K)
Gasoline 2.02 kJ/(kg * deg K)
Water 4.18 kJ/(kg * deg K)
Ethanol 2.43 kJ/(kg * deg K)
Methanol 2.51 kJ/(kg * deg K)

Fuel and other liquids also have what's called latent heat. This is the heat energy required to vaporize 1 kg of the liquid. The fuel in an internal combustion engine has to be vaporized and mixed thoroughly with the incoming air to produce power. Liquid gasoline does not burn. The energy to vaporize the fuel comes partially from the incoming air, cooling it. The latent heat energy required is actually much larger than the specific heat. That the energy comes from the incoming air can be easily seen on older carbureted cars, where frost can actually form on the intake manifold from the cooling of the charge.

The latent heat values of different liquids are shown here:

Gasoline 350 kJ/kg
Water 2256 kJ/kg
Ethanol 904 kJ/kg
Methanol 1109 kJ/kg

Most engines produce maximum power (with optimized ignition timing) at an air-fuel-ratio between 12 and 13. Let's assume the optimum is in the middle at 12.5. This means that for every kg of air, 0.08 kg of fuel is mixed in and vaporized. The vaporization of the fuel extracts 28 kJ of energy from the air charge. If the mixture has an air-fuel-ratio of 11 instead, the vaporization extracts 31.8 kJ instead. A difference of 3.8 kJ. Because air has a specific heat of about 1 kJ/kg*deg K, the air charge is only 3.8 C (or K) degrees cooler for the rich mixture compared to the optimum power mixture. This small difference has very little effect on knock or power output.

If instead of the richer mixture about 10% (by mass) of water would be injected in the intake charge (0.008 kg Water/kg air), the high latent heat of the water would cool the charge by 18 degrees, about 4 times the cooling effect of the richer mixture. The added fuel for the rich mixture can't burn because there is just not enough oxygen available. So it does not matter if fuel or water is added.

So where does the knock suppression of richer mixtures come from?

If the mixture gets ignited by the spark, a flame front spreads out from the spark plug. This burning mixture increases the pressure and temperature in the cylinder. At some time in the process the pressures and temperatures peak. The speed of the flame front is dependent on mixture density and AFR. A richer or leaner AFR than about 12-13 AFR burns slower. A denser mixture burns faster.

So with a turbo under boost the mixture density raises and results in a faster burning mixture. The closer the peak pressure is to TDC, the higher that peak pressure is, resulting in a high knock probability. Also there is less leverage on the crankshaft for the pressure to produce torque, and, therefore, less power.

Richening up the mixture results in a slower burn, moving the pressure peak later where there is more leverage, hence more torque. Also the pressure peak is lower at a later crank angle and the knock probability is reduced. The same effect can be achieved with an optimum power mixture and more ignition retard.

Optimum mix with "later" ignition can produce more power because more energy is released from the combustion of gasoline. Here's why: When hydrocarbons like gasoline combust, the burn process actually happens in multiple stages. First the gasoline molecules are broken up into hydrogen and

carbon. The hydrogen combines with oxygen from the air to form H2O (water) and the carbon molecules form CO. This process happens very fast at the front edge of the flame front. The second stage converts CO to CO2. This process is relatively slow and requires water molecules (from the first stage) for completion. If there is no more oxygen available (most of it consumed in the first stage), the second stage can't happen. But about 2/3 of the energy released from the burning of the carbon is released in the second stage. Therefore a richer mixture releases less energy, lowering peak pressures and temperatures, and produces less power. A secondary side effect is of course also a lowering of knock probability. It's like closing the throttle a little. A typical engine does not knock when running on part throttle because less energy and therefore lower pressures and temperatures are in the cylinder.
This is why running overly-rich mixtures can not only increase fuel consumption, but also cost power.

Appendix XIX: Acronyms

Engine management specifics

Acronym	**Description**
ABS	Antilock Braking System
AMM	Air Mass Meter
BDM	Background Debug Mode
CANBUS	Controller Area Network (Car Area Network)
CPS	Crankshaft Position Sensor
DI	Direct Ignition
DICE	Dashboard Integrated Control Electronics
ECM	Engine Control Module
ECU	Engine Control Unit
EDU	Electronic Display Unit
FPR	Fuel Pressure Regulator
FPT	Full Pressure Turbo
HOT	High Output Turbo
IAT	Intake Air Temperature
LPT	Light Pressure Turbo / Line Printer Terminal
MAF	Mass Air Flow
MAP	Manifold Absolute Pressure
OBD	On Board Diagnostics
RAM	Random Access Memory
ROM	Read Only Memory
RPM	Revolutions Per Minute
SFI	Synchronous Flash Interface (Production test interface)
SID	System Information Display
TPS	Throttle Position Sensor
TWICE	Theft Warning Integrated Central Electronics
VSS	Vehicle Security System
WOT	Wide Open Throttle