Free Arduino Boost box

By

GTinnovation

1.Introduction
2.Description
3.Tools and parts
4.Electronic schematics & diagrams
5.Software
6.Procedure
7.Testing
8.Vehicle installation
9.Acronyms
10.Disclaimer
11.Links

1.Introduction

Imagine that you own a 1.8 Tsi/Tfsi Engine which is equipped with a Maf* sensor and a turbocharger. How grate would it be if you could just give your engine a 25hp to 30hp boost without affecting the factory ecu* code? With some soldering skills and some parts that will not cost more then 70$ this can be possible.

For years the tuning industry is selling tuning boxes* for a variety of cars and brands. Tuning boxes are substitutes of software remaps for more performance in a diesel or petrol engines.

Of course software remaps are always better but sometimes people can’t afford them or are scared to proceed in altering the factory code.

So…Here it is a homemade tuning box that will save you money and give you some power without affecting the software inside the ecu and can be installed or removed in seconds.

This box has been tested on a 1.8Tsi engines for about a year without problems. We have developed it for testing reasons and to give the client a small idea of how good the car will be if we remap the stock ecu but since the global economy is suffering we decided to give a piece of our mind for free ☺

With some or no modifications in the code it could work on 2.0ltfsi too but I have not tested that since we always remap those cars.

Feel free to email us with any questions / suggestions at :

info@gt-innovation.gr

2.Description

Using the well known Arduino* platform we developed this simple box without spending more then 70$ for the complete set of parts. Our main goal was to keep a low cost while having a trustworthy device.

So we used the Arduino* platform and some connectors and cables to interfere with the factory map sensor get the actual signal, alter it and then send it back to the Ecu*.

In most cases if you alter the signal too much factory Ecu* will understand that and will put your car in a safe mode. Of course once you reset it with a diagnostic and remove the boost box this fault code will not be displayed any more. After some tests we figured out how you could surpass that problem and keep the power in good levels.

What is this box? This is an electronic device that tricks the factory engine management computer in order to produce more power by adding boost. The boost signal is coming from the map* sensor and that is what we have to alter.

While you do tests use quality fuel. In Europe we use 95, 97 or 100 octane fuel so we suggest 100 octane or else you will need to decrease the number signal modification number thus you will have less power from the Arduino* boost box.

To sum up this will be a step by step guide to help you build your own tuning box for your 1.8tsi 160hp engine. I will try to be as analytic as i can.

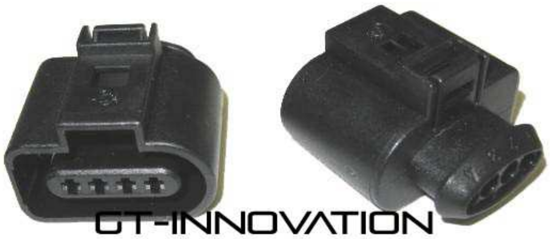
3.Tools And Parts

Tools: 1. Professional Soldering iron.
2 .Pc with winows xp or newer.
3. Usb mini to pc usb cable(usually inside the arduino).
4. Usb to printer usb cable.
5. Patience.
6. Vag com Cable (Vcds for testing purposes).
7. Multimeter

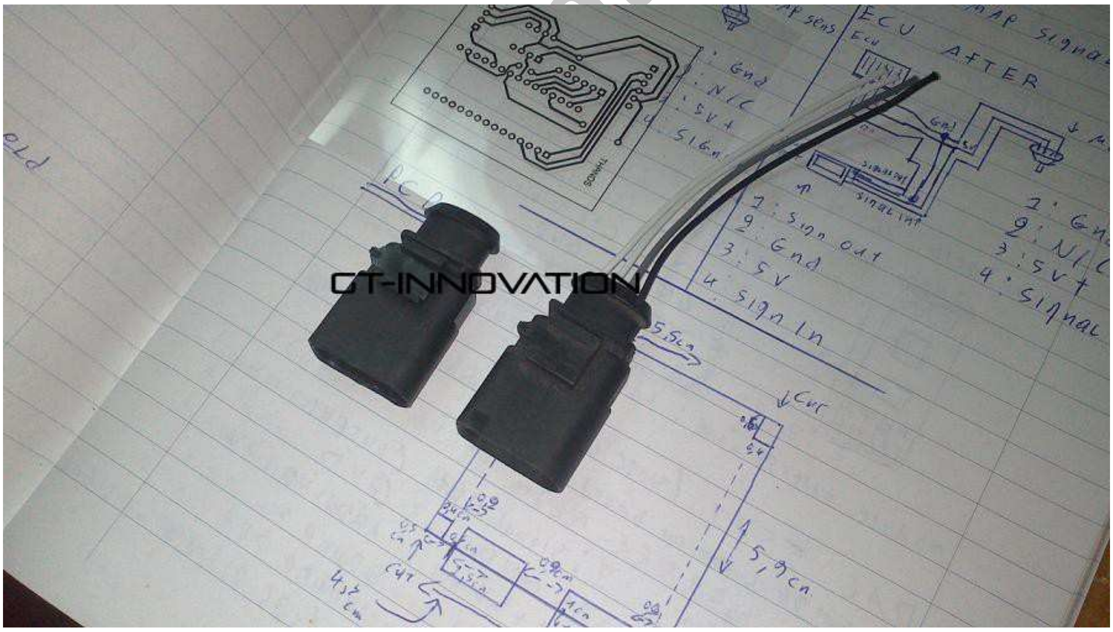
Parts: 1. Arduino nano 328p.
2. Small cables & long cables 4 different colors. (Red – Black – Green – Yellow)
3. Tekam enclosure 11.9 (or any other enclosure you like).
4. Old map sensor plug Female (you can cut from a bad harness) bosch code 1J0 973 704
5. Broken or damaged map sensor 2550mbar Bosch (1.8t 20v or any other 4 pin vw map sensor) or you can get a narrowband Bosch lambda female connector (4 wire) tyco 1k0 973 804 .
6. Dac* 5571 (Texas instruments) sot23 8bit 5v .You can order a sample from Texas instruments site for free, ☺ or you can buy on ebay.
7. Prototype pcb (piece 5.5cm x 5.9cm).
9. Sot23 8pin breakout board from ebay (or 6 pin breakout board for sot23 6 pin).
10. Atmega* 328 with 1280 or 2560 Arduino board(for boot loader replacement) or Arduino duemilanove (my personal choice).

Parts Photo

Female connector

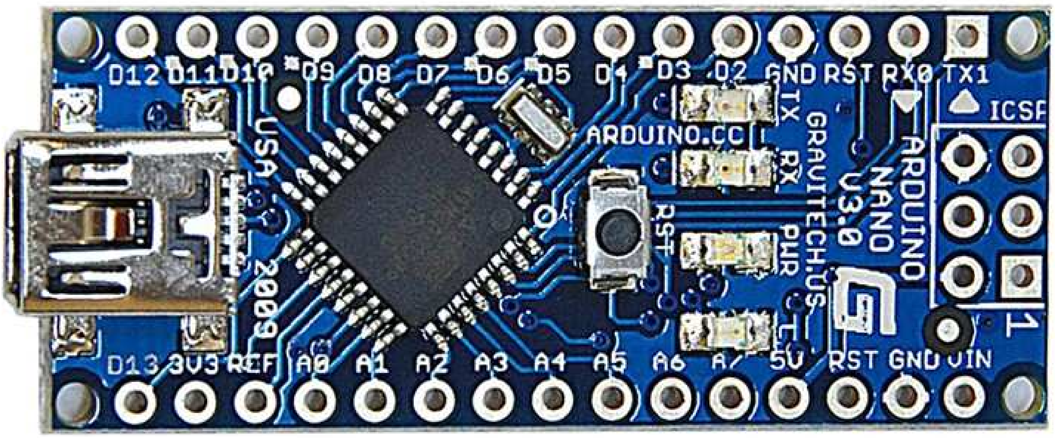


Male connector



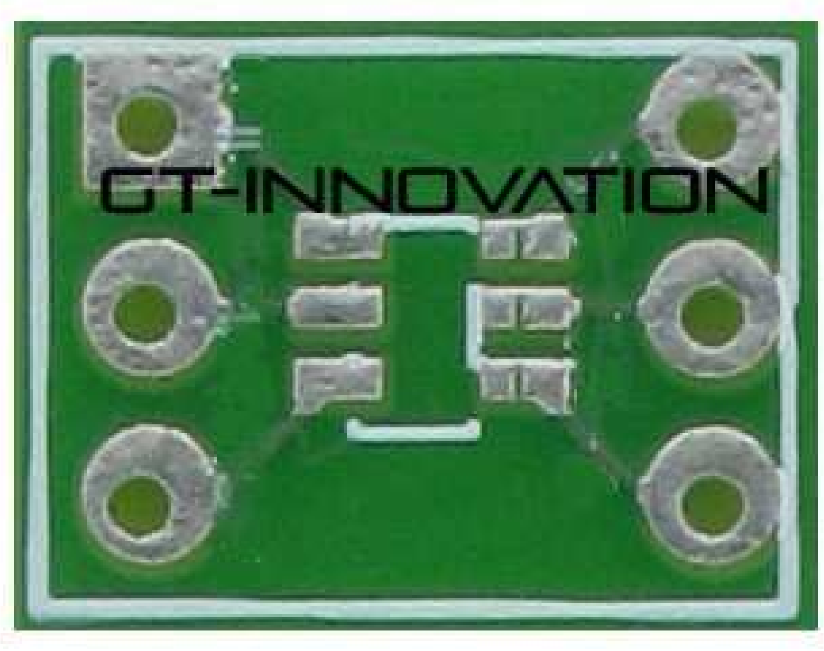
Map sensor plug source : http://kyae.en.alibaba.com/product/518465271-212848327/Bosch_VW_4_poles_Male_MAP_sensor_connectors.html

Arduino nano v3

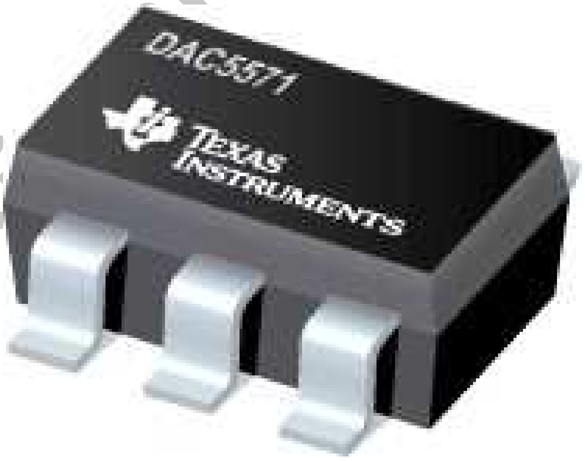


Teko Tekam Enclosure

Sot23 6pin pcb

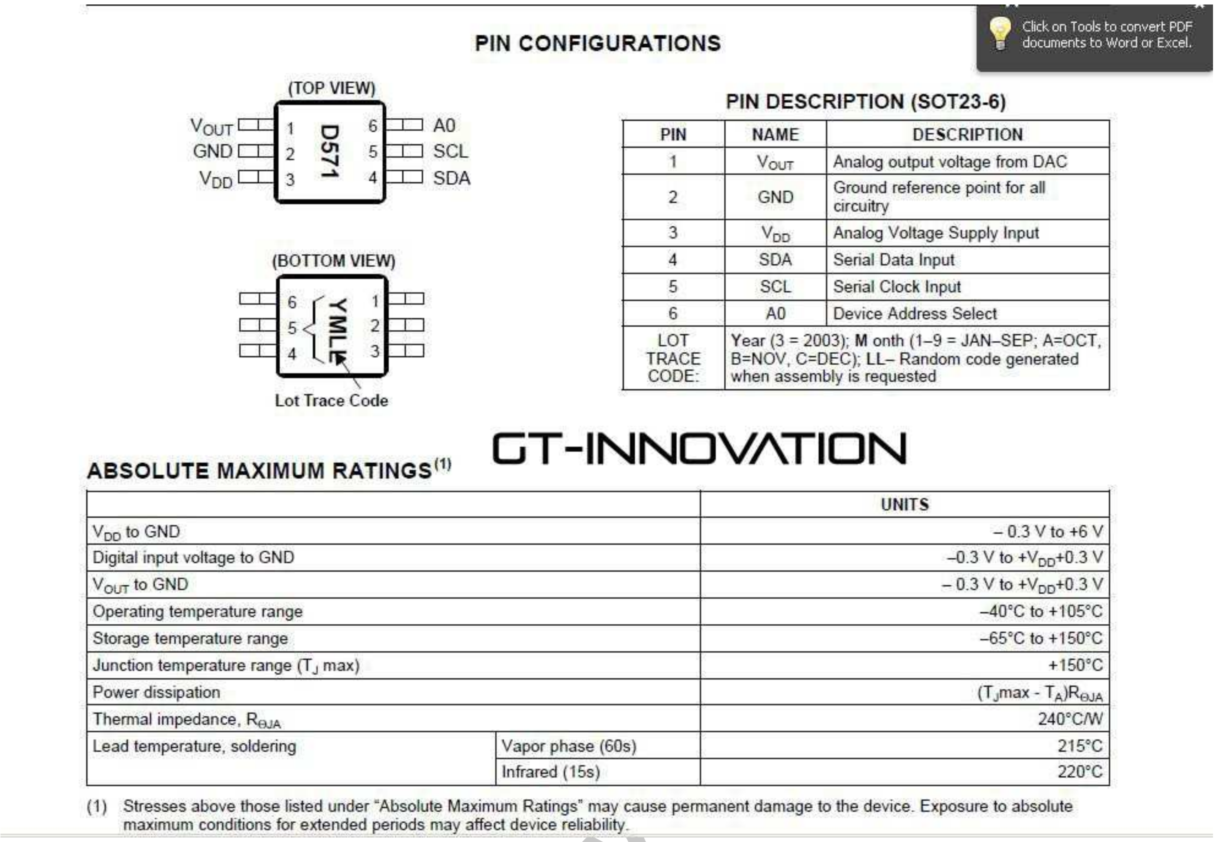


Dac 5571 from Texas Instruments



4.Electronic Schematics

Dac5571 diagram



PIN CONFIGURATIONS

PIN DESCRIPTION (SOT23-6)

PIN	NAME	DESCRIPTION
1	V_{OUT}	Analog output voltage from DAC
2	GND	Ground reference point for all circuitry
3	V_{DD}	Analog Voltage Supply Input
4	SDA	Serial Data Input
5	SCL	Serial Clock Input
6	A0	Device Address Select
LOT TRACE CODE:	Year (3 = 2003); M onth (1–9 = JAN–SEP; A=OCT, B=NOV, C=DEC); LL– Random code generated when assembly is requested	

ABSOLUTE MAXIMUM RATINGS[1]

		UNITS
V_{DD} to GND		– 0.3 V to +6 V
Digital input voltage to GND		–0.3 V to +V_{DD}+0.3 V
V_{OUT} to GND		– 0.3 V to +V_{DD}+0.3 V
Operating temperature range		–40°C to +105°C
Storage temperature range		–65°C to +150°C
Junction temperature range (T_J max)		+150°C
Power dissipation		($T_J$max - T_A)$R_{\theta JA}$
Thermal impedance, $R_{\theta JA}$		240°C/W
Lead temperature, soldering	Vapor phase (60s)	215°C
	Infrared (15s)	220°C

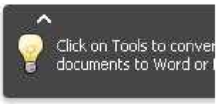
[1] Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum conditions for extended periods may affect device reliability.

Pinout to arduino :

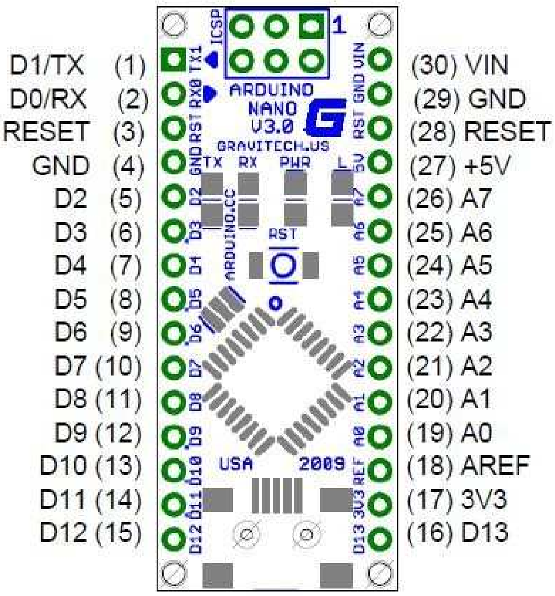
1.Signal output to ecu harness
2.Ground (gnd)
3.+5v
4.Scl (arduino Scl)
5.Sda (arduino Sda)
6.Address (i2c Address usually you can connect to gnd check the diagram of the pcb)

Arduino nano Diagram

Arduino Nano Pin Layout

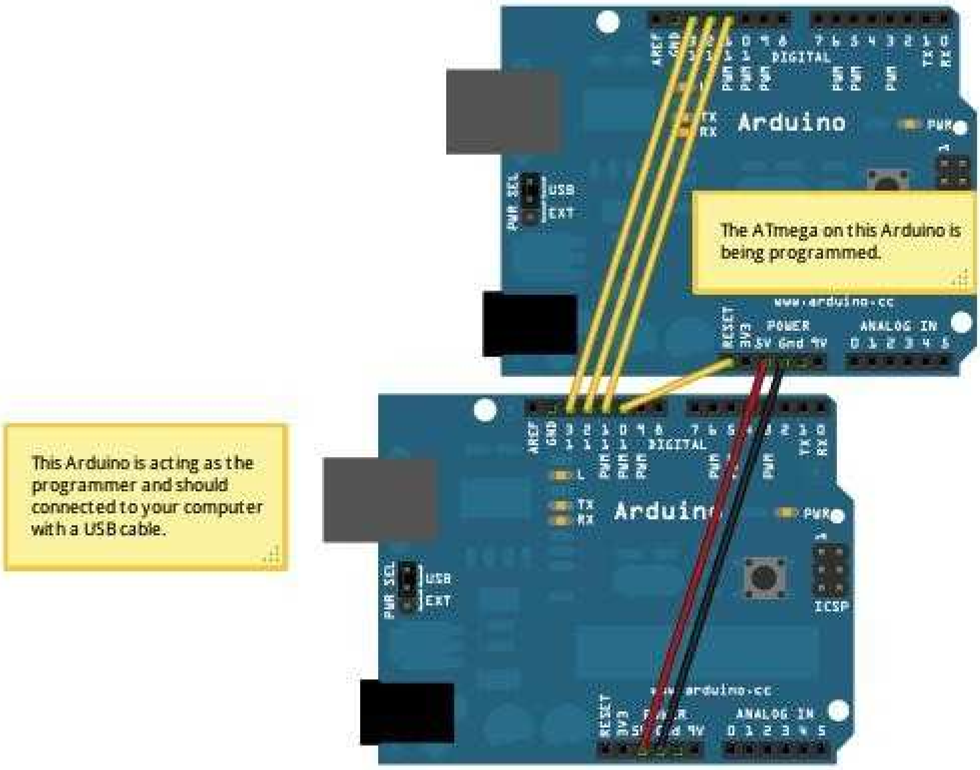


Click on Tools to conver
documents to Word or



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A0-A7	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Arduino ISP connection

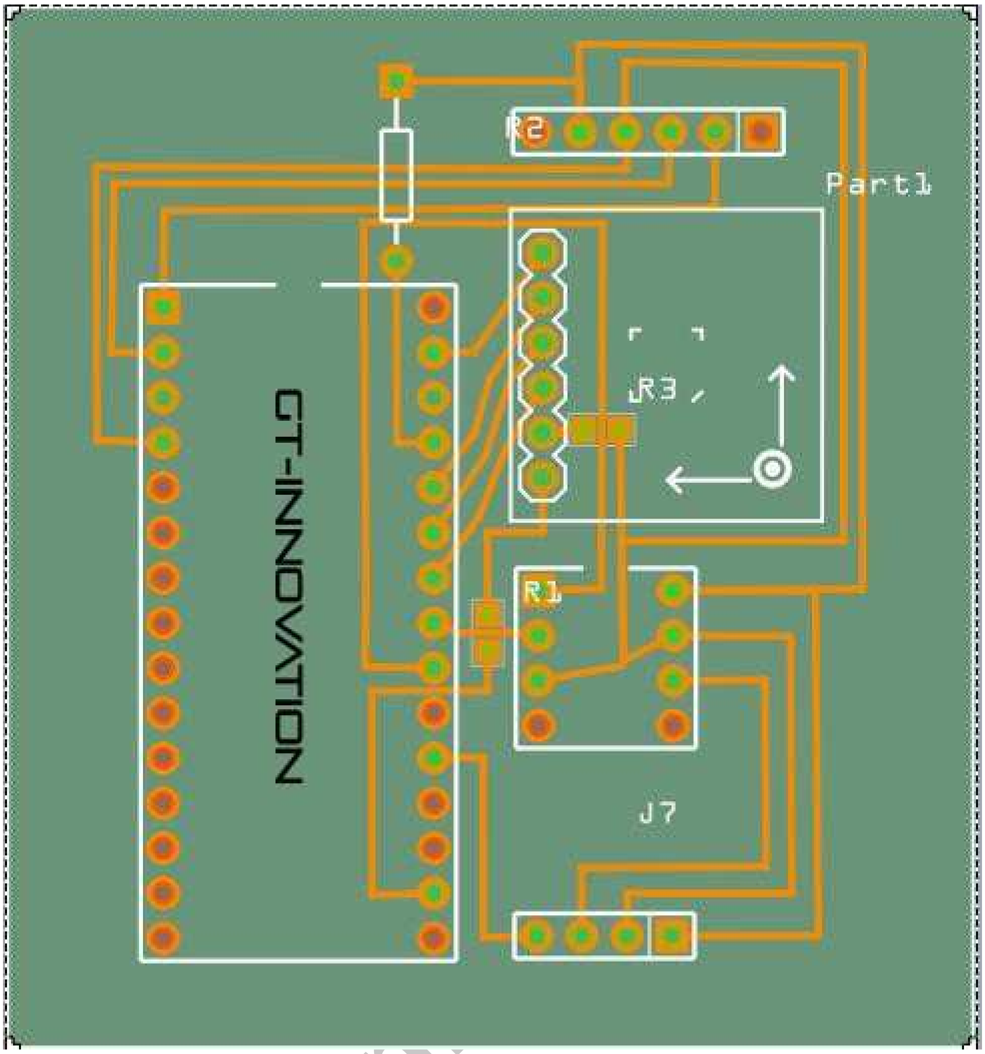


The ATmega on this Arduino is being programmed.

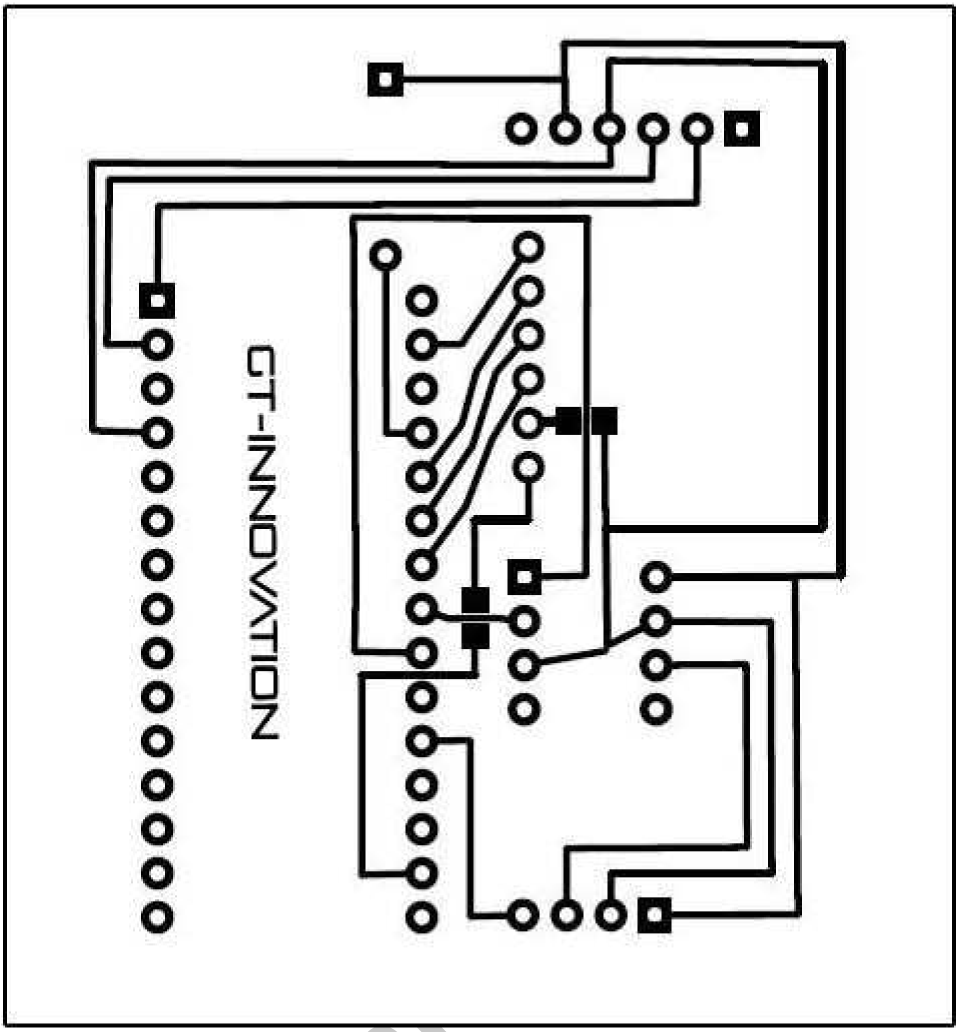
This Arduino is acting as the programmer and should connected to your computer with a USB cable.

Pcb Layers and layouts.

Now in the next photo you will see a pcb board containing the parts and the connections. You will notice that there is an area with another part that is missing above the Dac5571.This are was left as it is with the connections open for a feature accelerometer installation (Adxl335).You will not need it right now so leave it as it is. You will find it also in the fritzing pcb design in the download section. There is also a resistor part but actually you will use a bypass wire on that.There is no need for any resistors in the design.



The usb port of Arduino should point to the bottom of the pcb. I have not included a pin to pin diagram connection because from the way the pcb is arranged it will not give you any more information. But you can see it by opening the fritzing pcb *.fzz file.



The output to the map sensor connector is from left to right 1 2 3 4 . Remember to think of the above pcb as a transparent bottom view.

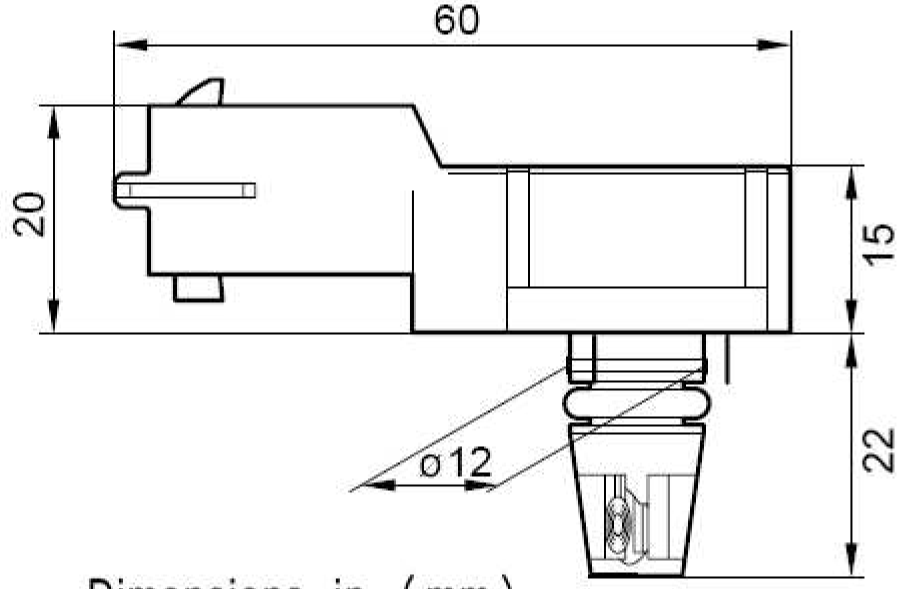
1.Yellow(signal from map sensor connector female plug) input
2.Green(signal for the map sensor connector male plug) output
3.Black(Gnd from map sensor connector male plug to female plug and to pcb)
4.Red(+5v From map sensor connector male plug to female plug and to pcb)

The pcb is using power from the ecu that is why you need to wire both plugs and the pcb to have a working circuit.Only the green and yellow wires are different.

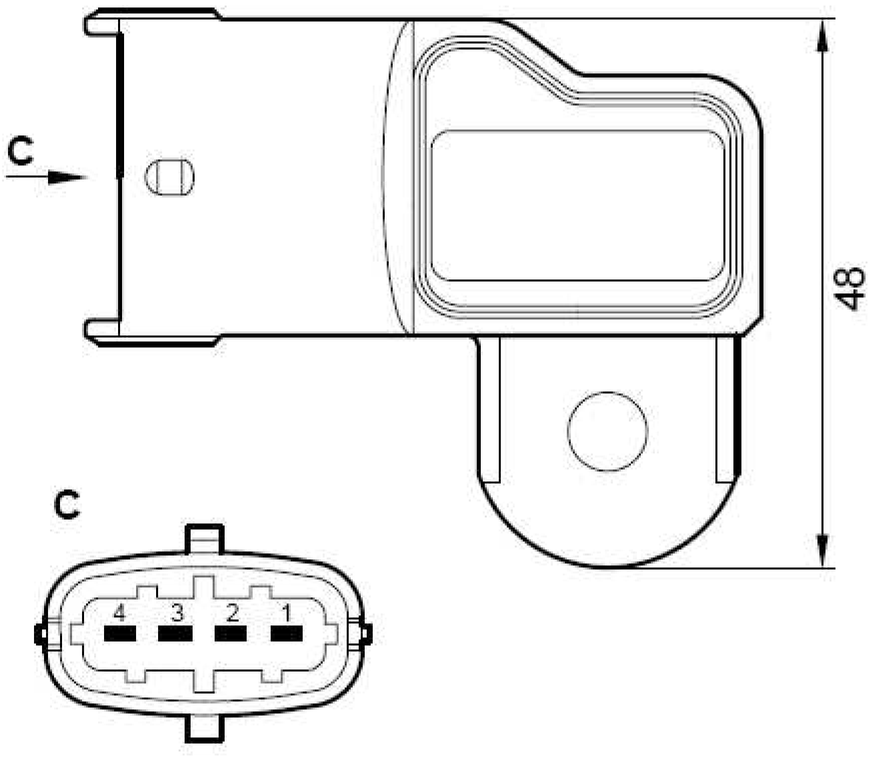
And some photos of a finished prototype with some mistakes corrected on the fly.

Connector-pin assignment

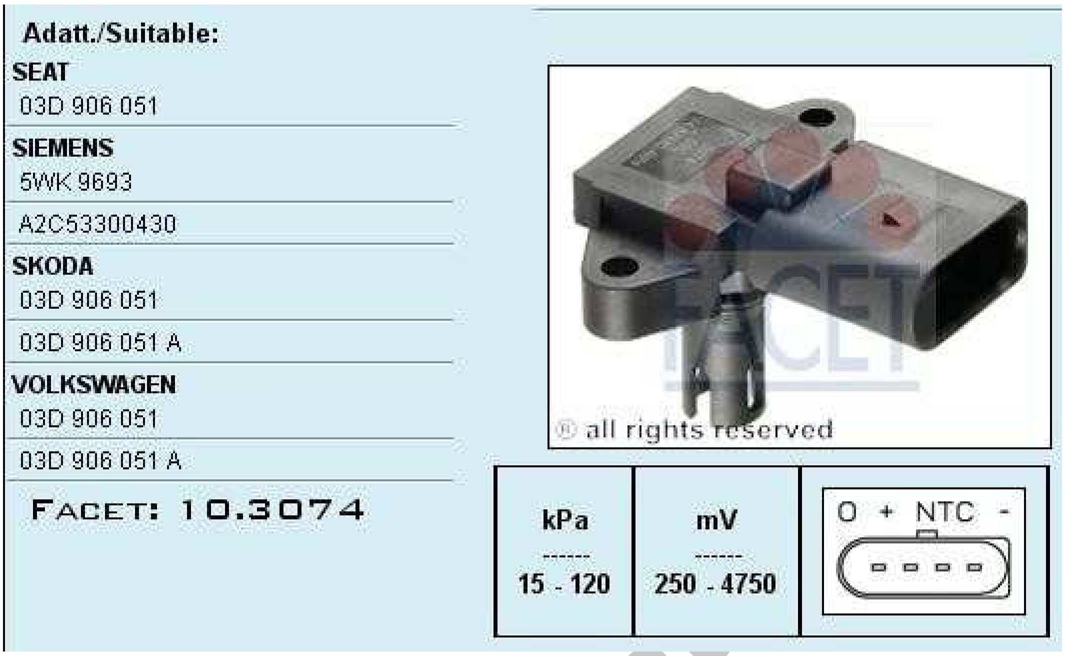
Pin 1	Ground
Pin 2	NTC resistor
Pin 3	+5 V
Pin 4	Output signal



Dimensions in (mm)



You do not need to connect the pin 2 since there is no wire coming from the ecu.



Adatt./Suitable:

SEAT
03D 906 051

SIEMENS
5WK 9693
A2C53300430

SKODA
03D 906 051
03D 906 051 A

VOLKSWAGEN
03D 906 051
03D 906 051 A

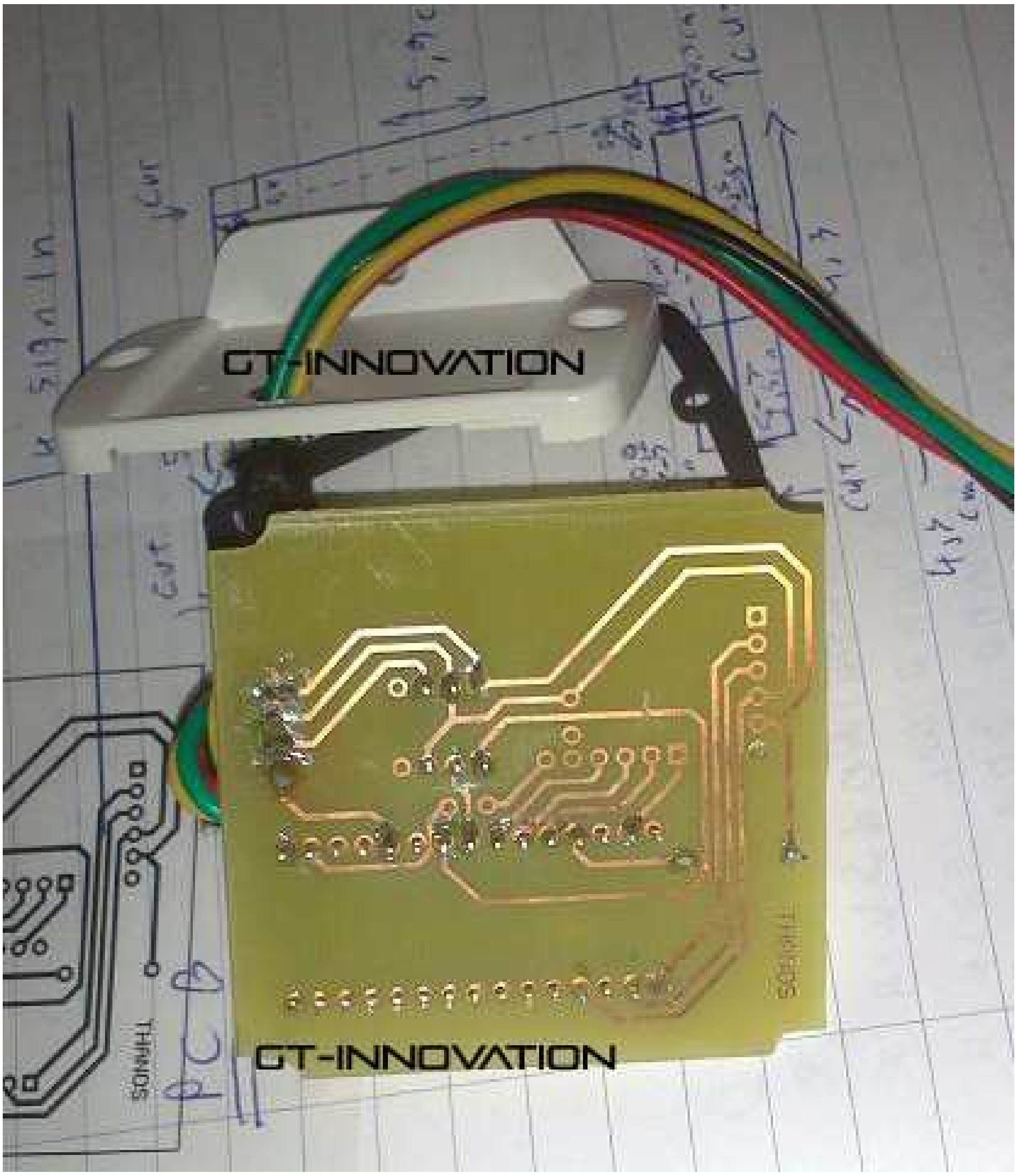
FACET: 10.3074

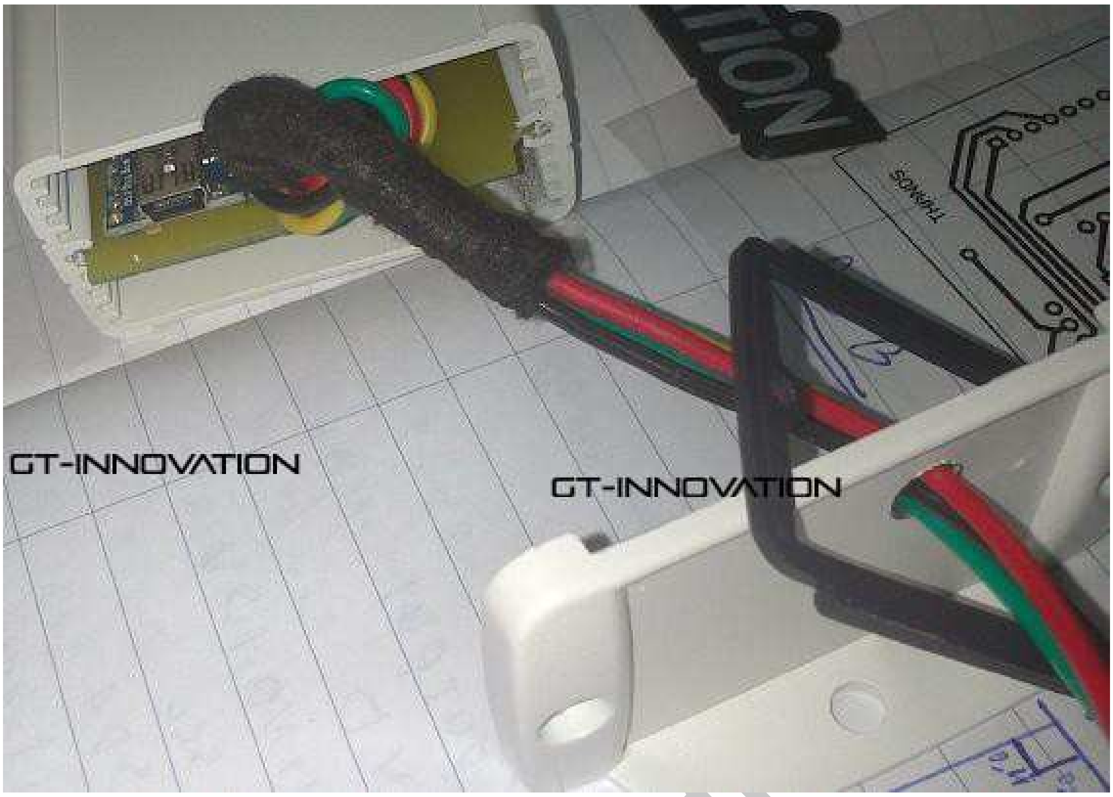
® all rights reserved

kPa	mV
------	------
15 - 120	250 - 4750

This is the actual connector which you will see on your car. From right to left 1.2.3.4 as the above picture

cut
PCB


GT-INNOVATION
GT-INNOVATION


GT-INNOVATION
GT-INNOVATION

GT-INNOVATION
PCB

5.Software

1.Windows pc
2.Arduino version 0022(cause I used that for the software)
3.Arduino version 1 (cause I used that one to make avr isp and load the new bootloader)
4.Hyperterminal(For communication with the arduino) mostly for testing
5.Optiboot atmega328 or optiboot atmega168 for quicker arduino boot time.
6.Boost box simple . pde (original boost box code).
7.Wire library for arduino(for dac5571 communication).
8.Vag com software(vcds for testing purposes).
9.Fritzing freeware pcb fabrication software.

Download Links:

http://www.gt-innovation.gr/downloads/arduino-1.0.rar

http://www.gt-innovation.gr/downloads/arduino-0022.rar

http://www.gt-innovation.gr/downloads/Boost_Box_Simple.rar http://www.gt-innovation.gr/downloads/bootloader.rar

http://www.gt-innovation.gr/downloads/fritzing.2013.02.25.pc.zip

And the pcb schematic and connection diagram here in fritzing format and the pcb in pdf :

http://www.gt-innovation.gr/downloads/Tsi-Boost-fritzing.rar

http://www.gt-innovation.gr/downloads/Tsi-Boost_etch_copper_bottom.rar

6.Procedure

Bootloader flashing :

If you don`t change the original boot loader arduino will not boot in time and after some ignitions you wil get a dtc for the map sensor. Ecu will think that the sensor is missing because it does not get any signal until the Arduino starts up.

So the procedure for programming the Optiboot boot loader is the following

To use your Arduino board to burn a boot loader onto an AVR, you need to follow a few simple steps.

1. Open the **Arduino ISP** firmware (in **Examples**) to your Arduino board.
2. **Note for Arduino 1.0:** you need to make one small change to the Arduino ISP* code. Find the line in the heartbeat() function that says "delay(40);" and change it to "delay(20);". <I never did that>
3. Select the items in the **Tools > Board** and **Serial Port** menus that correspond to the board you are using as the programmer (not the board being programmed).
4. Upload the ArduinoISP sketch.
5. Wire your Arduino board to the target as shown in the diagram below. (**Note for the Arduino Uno:** you'll need to add a 10 uF capacitor between reset and ground.)
6. Select the item in the **Tools > Board** menu that corresponds to the board **on which you want to burn the boot loader** (not the board that you're using as the programmer). See the board descriptions on the environment page for details.
7. Use the **Burn Bootloader > Arduino as ISP*** command.

Now if you use my files you will not need to alter any configuration or add any library to the Arduino compiler.I have already made the configuration for flashing the nano 328 with the appropriate bootloader. You will just need to use Arduino 0022 or flashing the code and 1.0 for burning the boot loader into the nano 328.

In the 022 Version under tools -> Board you will find a board named Arduino Nano ATmega328 optiboot and Arduino Nano ATmega168 optiboot .

Use this board so you can flash the code into the arduino nano.
In arduino 1.0 version you will find the Arduino Nano ATmega328 optiboot only since I did not used the 168 for flashing back the original bootloader.

Here are the complete notes of what I did and why:

Change boot loader due to optiboot or lady ada boot fix.

```
000567 - Manifold Pressure / Boost Sensor (G31): Signal too Low
            P0237 - 000 -  -  - Intermittent - MIL ON
```

to fix the boot loader issue i did the following

Rename optiboot_atmega328.hex boot loader to ATmegaBOOT_168_atmega328.hex in the Atmega folder inside the version 1 folders C:\arduino-1.0\hardware\arduino\bootloaders\atmega

Then i programmed Arduino duemilanove to arduino avr isp and chose Atmega 328 nano choice in the devices
on Arduino Gui.
then i loaded the Arduino boot loader using the Duemilanove as isp* programmer with version 1 code.

After that i opened the 0022 folder C:\arduino-0022\hardware\arduino

and then i wrote the following to C:\arduino-0022\hardware\arduino\boards.txt

```
############################################################
###

optiatmega328.name=Arduino Nano ATmega328 optiboot

optiatmega328.upload.protocol=stk500
optiatmega328.upload.maximum_size=30720
optiatmega328.upload.speed=115200

optiatmega328.bootloader.low_fuses=0xFF
optiatmega328.bootloader.high_fuses=0xDE
optiatmega328.bootloader.extended_fuses=0x05
optiatmega328.bootloader.path=atmega
optiatmega328.bootloader.file=optiboot_atmega328.hex
optiatmega328.bootloader.unlock_bits=0x3F
optiatmega328.bootloader.lock_bits=0x0F

optiatmega328.build.mcu=atmega328p
optiatmega328.build.f_cpu=16000000L
optiatmega328.build.core=arduino
```

then i opened 022 Arduino gui and chose atmega 328 optiboot and compiled and uploaded the sketches.
Once you do that you have completely finished with the coding part of the device and you are almost ready to test.
While testing the power that we could achieve from that device we found out that we should not exceed the number 9 in this line of the arduino sketch code.

```
int ScrB = 9; //Powertune value for voltage reduction
```

For instance with a value of 10 the car will jerk a lot until the ecu regulates the boost.

The value of 9 should be used with premium 98 or 100 octane fuel. With 7 or 8 you are safe for 91 to 95 octane fuel. The boost will be around 1.1 bar using a value of 9 while the stock software does not exceed 0.8 bar.

7.Testing

There are a lot of ways to test your device but the basic one is to see if the signal from and to the device is ok. So using a multimeter check the voltage of the output signal while your arduino is connected to the usb port of your computer. Since power is coming from the ecu power output for the sensor and now you are not connected there, you will need to power up your arduino using your pc. Power on the red and black cable should be around 4.5x Volt .The value should be 1.6xx volt on the output (green cable) because we have a constant voltage variable in the code before we start getting signal :

```
int sensorValue = 400;  // variable to store the value coming from
the sensor
```

Now this device was never tested on any kind of altitude so I do not know how good it will work on 500+ meters from the sea.
The code is regulating after some pressure to avoid any kind of altitude problems and to avoid boost spikes on wot.

```
   if(DACval > 130)                    // if more then 1300 mbar
reduce i2c dac value for scrambling
   {
    percent = (ScrB / 10) * (DACval / 4);// percentage calculation
    DACval = DACval - percent;          // value reducer
   }
```

This code is what makes the device work so you can tweak it but I do not recommend it.

Next test requires a Vcds unit and will be performed after the successful build of the device. This test is just for vehicle health and should be performed if you need to be sure that everything works fine apart from the electronics you build.

Connect vag com to the obd port and go into the engine section.
After that locate the advanced measurement blocks and find the knock sensor blocks. You can log them and see if the knock signal is getting much higher then the stock one by comparing before and after wot tests for at least 2 or 3 gear changes.

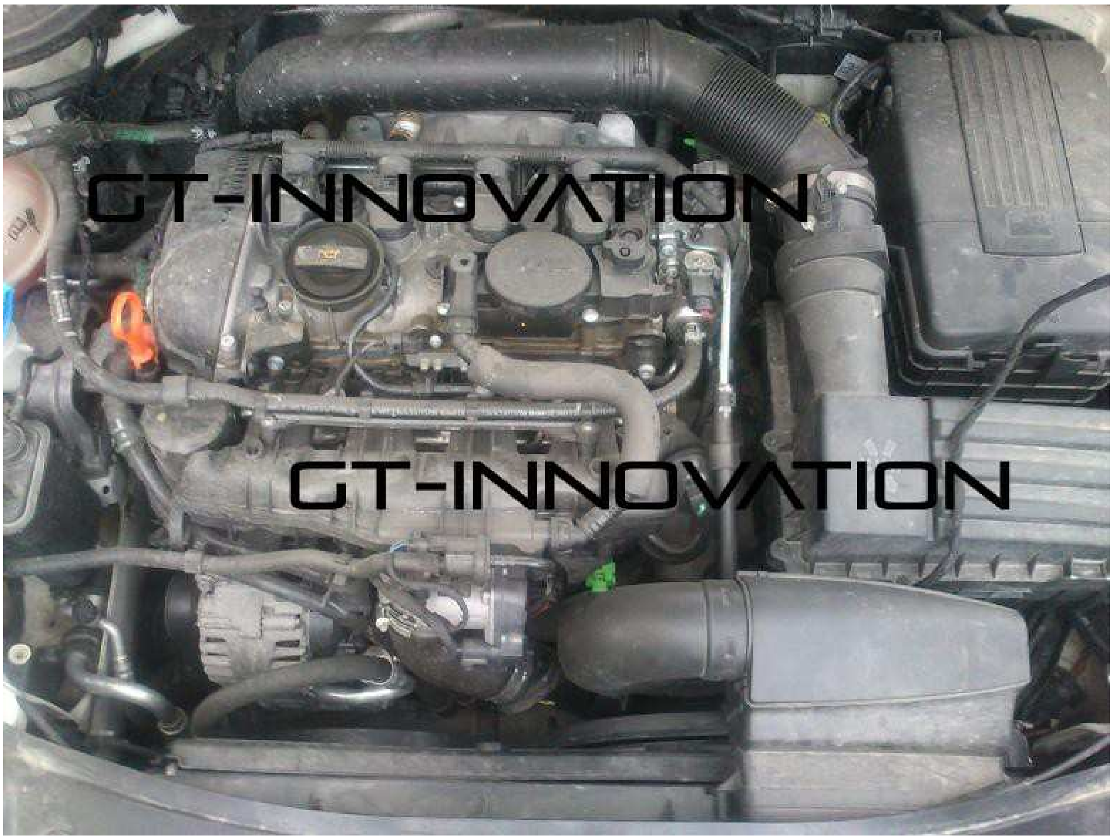
This device was tested on 3 brands with the same engine described below:

Audi A3 1.8 Tsi 160hp
Seat leon 1.8 Tsi 160hp
Skoda Octavia 1.8 Tsi 160hp

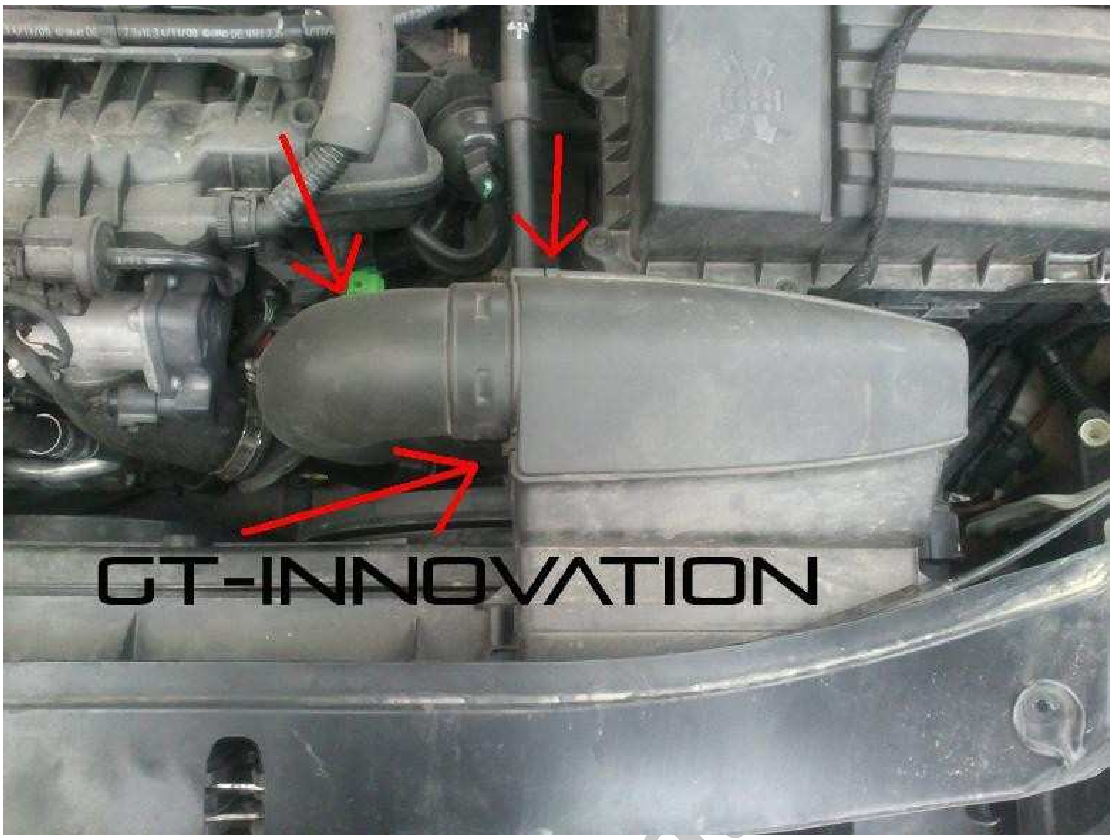
In some cases this device produced 35 to 40 hp due to good fuel and cold air.

8.Vehicle installation

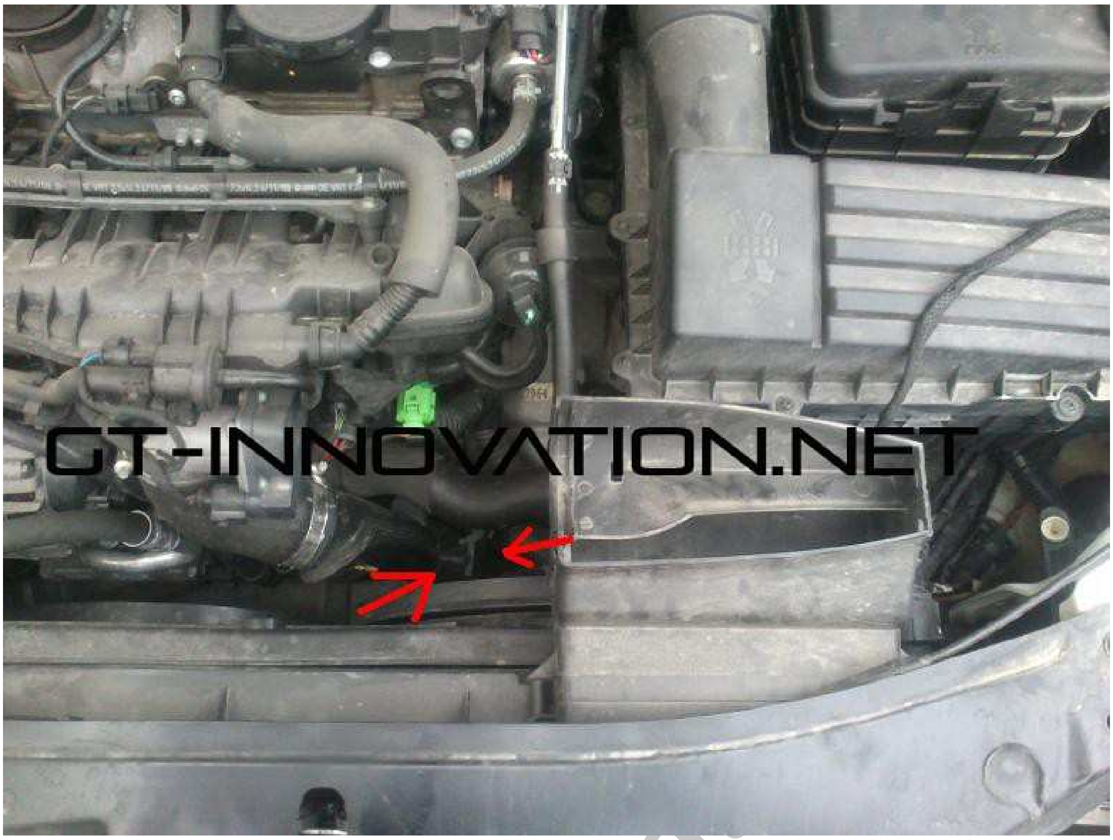
The installation will be easy if you let enough cable from the sensor connectors to the main board. This is your engine bay.

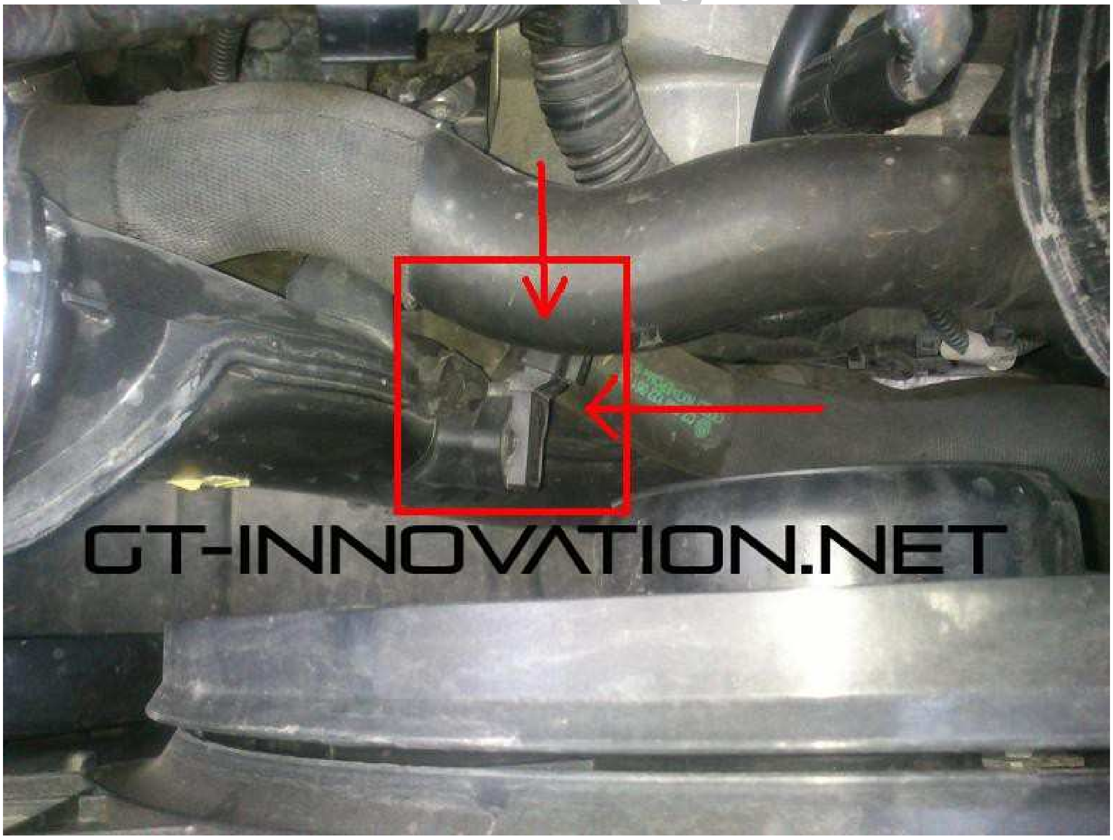


Remove the plastic cold air intake cover and that plastic pipe that is connected to it.



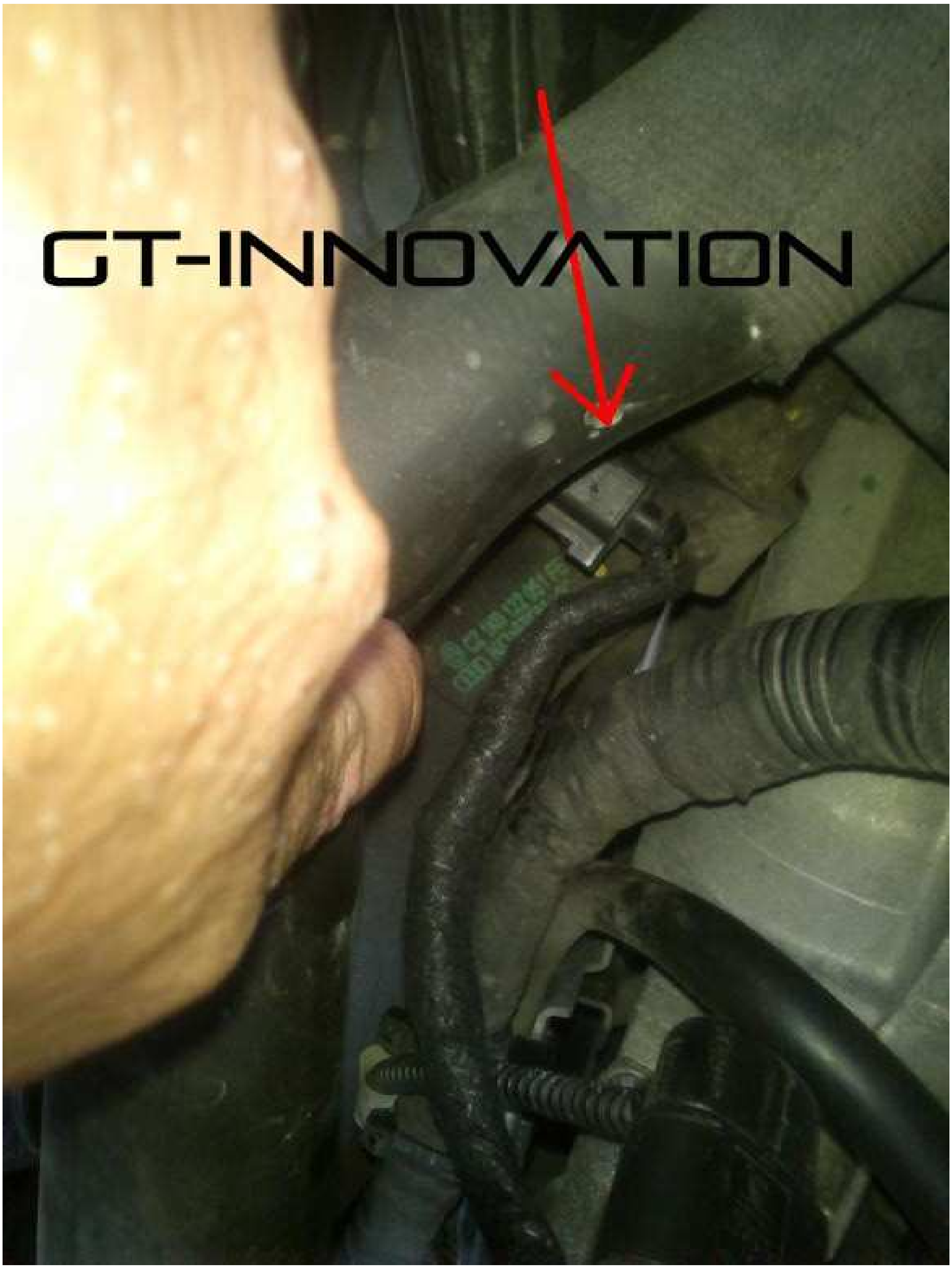
After that you will need to locate the map sensor. The map sensor is located on the intake manifold pipe just before the throttle.
Check between the radiator ant the engine intake with your right hand and check under the big black radiator pipe.





Make sure you have a cold engine because otherwise you will get some nasty burns.

Now remove the connector and install the custom boost box between the map sensor and the map sensor harness



Once you are finished you can hide the box inside the battery box assuming you left enough cable. Ready for test driving and have fun.

9.Acronyms

Tuning Box : Electronic device that interrupts the factory harness and alters the sensor data from the sensor to the ecu(aka piggy back).
Ecu : Engine management computer(Electronic control unit)
Maf : Mass Air flow sensor
Map : Manifold Air pressure sensor
Arduino : Programmable microcontroller circuit using an open source project.
Dac : Digital to analog converter
Atmega : Microcontroller type from Atmel
ISP : In-System Programmer

10.Disclaimer

Gt-innovation is not responsible for any damage or misuse of this document. Use it at your own risk.
All Rights Reserved. All Content Copyright and other rights reserved by its Respective Owners. No Content May Be Duplicated Without Express Written Consent.

11.Links

Arduino source : http://arduino.cc/en/
Texas instruments : http://www.ti.com/
Teko Tekam : http://www.teko.it/
Fritzing pcb design : http://fritzing.org/
Vag com : http://www.ross-tech.com/
Gt-innovation : http://www.gt-innovation.net/