

Kompressionstester bauen (DIY)

Weil mein Post zum Thema recht große Resonanz gefunden hat, hier nun die offizielle Anleitung zum Bau eines DIY Kompressionstesters. Das ganze beruht auf einer Anleitung die ich selbst im Netz gefunden habe:

<http://www.instructables.com/i...ngine-Compression-Tester/>

Der Eintrag ist von John Doss, dem Kopf hinter TwistedRotors die in den USA verschiedene [Kompressionstester](#) für Wankelmotoren gebaut und verkauft haben.

Die Idee ist einen Drucksensor mit Adaptern so zu verbauen, dass er in das Loch einer Zündkerze geschraubt werden kann. Der Sensor wird an ein arduino developer board angeschlossen in dem das analoge Spannungssignal des Sensors digitalisiert wird und aus der Frequenz der Kompressionspulsen die Drehzahl errechnet wird. Dieses Signal muss dann über ein Serial to USB Kabel an einen Computer übertragen werden, auf dem dann die grafische Ausgabe erfolgt.

Hardware (ein nicht unerheblicher Teil der Arbeit besteht im Beschaffen der erforderlichen Hardware):

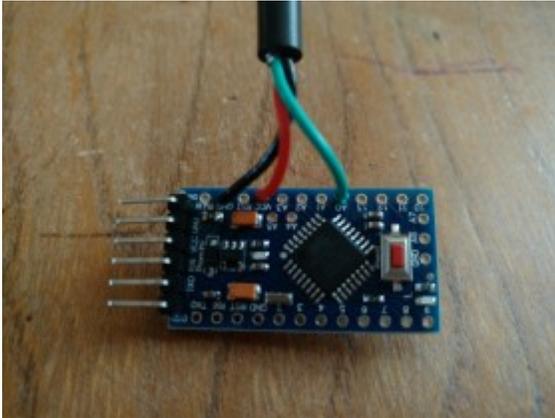
Was man benötigt:

- Arduino Pro Mini 328P 5V: <https://www.ebay.de/itm/ATmega...8e8784:g:QQIAAOSw-0xYinKq>. So ein Board bekommt man auch für weniger als 2€ aus China, in dem Fall muss man aber recht lange darauf warten: <https://www.ebay.de/itm/Pro-Mi...6829cd:g:9BsAAOSwNsdXTTOo>
- Ein FTDI FT232RL USB to UART(TTL) Serial Kabel: <https://www.ebay.de/itm/USB-TO...ksid=p2060353.m2749.l2649>
- Einen 200psi Drucksensor: <https://www.ebay.de/itm/Drucks...ksid=p2060353.m2749.l2649>, habe ich nur in China gefunden
- Einen Adapter 1/8" NPT auf M14x1,25: <https://www.kosoeurope.com/zub...8x28-m14x1-25x15mm?c=3132> John Doss macht das mit einem 1/8" NPT auf 1/4" NPT Adapter, ich habe mir allerdings bestätigen lassen, dass das Gewinde ein M14x1,25 ist. Da man die Verbindung nicht mehr trennen will, geht es vielleicht auch mit dem 1/4" NPT gut, welches dann konisch dichtet.
- Einen [Zündkerzen](#)-Adapter von Dorman, gibt es in kurz <https://www.rockauto.com/de/moreinfo.php?pk=14578&jsn=3> oder länger <https://www.rockauto.com/de/moreinfo.php?pk=14580&jsn=8>, welchen ihr benötigt hängt davon ab, welchen 1/8" NPT auf M14x1,25 Adapter ihr benutzt, bzw. lang dessen Gewinde ist. Für den von mir angegebenen Adapter von Koso braucht ihr die lange Variante 42004.
- Einen O-Ring 11,3x2,4mm: <https://www.ebay.de/itm/10-er-...ksid=p2060353.m2749.l2649>
- Ein Gehäuse für die Elektronik: <https://www.ebay.de/itm/Kunsts...ksid=p2060353.m2749.l2649>
- Gewindedichtung, ich habe Loctite 542 genommen: <https://www.ebay.de/itm/LOCTIT...7c4a30:g:alwAAOSwtj1a6rJM> sollte aber auch mit Teflonband gehen, oder anderen Dichtmitteln Eurer Wahl

Hardware bauen:

Die Arduino Boards kommen in der Regel mit 3 Stiftleisten, von denen wir nur die rechtwinklige benötigen, die wird auf die entsprechende Seite aufgelötet, ausserdem werden die 3 Kabel vom Sensor auf die Platine

gelötet



Bei allen Sensoren die ich in den Händen hatte ist das rote Kabel VCC (5V DC), das schwarze GND und das grüne der Ausgang Vout der auf den A0 Anschluss gelötet werden muss. Ich habe alle Lötstellen auf der Platinenrückseite verlötet, insgesamt sind es 8 Stück.

Jetzt Kann das USB Kabel an das Board angeschlossen werden (schaut auf die Pinbelegung auf dem Board, die steht drauf und zeigt Euch wie rum das Kabel gesteckt werden muss), das ganze kommt dann in das Gehäuse:



Das hält so schon gut, wenn man sicher gehen möchte kann man die Kabel jeweils 180° umschlagen und mit 2 Kabelbinden das Gehäuse und Kabel fixieren, damit hat man auch eine Zugentlastung.

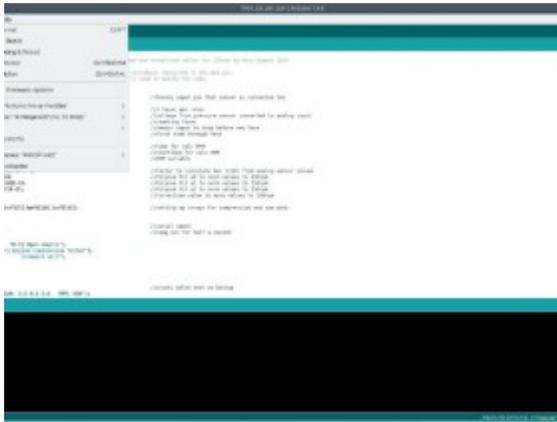
Das Kabel vom Sensor trennen und den Sensor mit dem Dichtmittel der Wahl in den 1/8" NPT Adapter schrauben und diesen dann (mit Dichtmittel) in den Dorman Zündkerzenadapter. Auf den Dorman Adapter kommt noch der O-Ring und ihr seid fertig mit der Hardware.



Software installieren:

Ihr braucht die Arduino IDE, gibt es kostenlos zum runter laden hier: <https://www.arduino.cc/en/main/software>
Das Programm muss installiert werden, hat bei mir unter Windows 7, Windows 10 und Arch Linux funktioniert, wobei ich bei einem Windows 10 Tablet Probleme mit den Treibern hatte, das hat später das Board nicht erkannt.

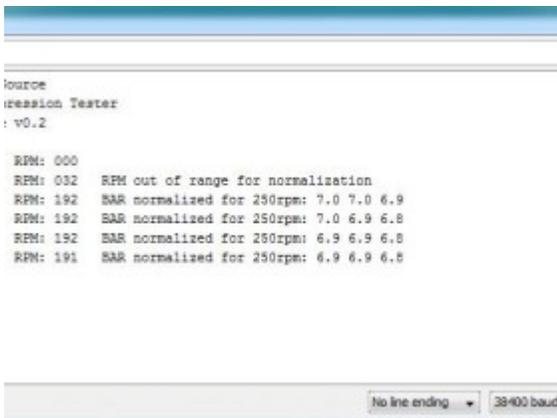
Nachdem die Software installiert ist könnt ihr das Programm TR01_OS_vxx_bar.zip entpacken und in der Arduino IDE öffnen.



Unter "Tools" muss das Board gewählt werden (Arduino Pro or Pro Mini) unter Prozessor Atmega 328P (5V, 16MHz). Wichtig ist den richtigen [Port](#) zu wählen (unter Windows COMx), wenn ihr nicht sicher seid welcher der richtige ist, probiert alle durch und klickt auf "Get Board Info". Wenn da was kommt ist es der richtige [Port](#). Nun kann man im Reiter "Sketch" auf "Upload" klicken und unter "Tools" den "Serial Monitor" öffnen. Im Serial Monitor muss die Baut-Rate auf 57600 gestellt werden. Ihr bekommt dann die Nachricht:

TR-01 Open-Source
Rotary Engine Compression Tester
[Firmware](#) Vxvyx

Wenn ihr das seht, kann es los gehen. Zündkerze rausschrauben, Exzenterwellensensor abstecken, Sensor ins Zündkerzenloch schrauben und den vorher warm gefahrene Motor ein paar Sekunden drehen lassen. Die Ausgabe sieht dann so aus:



Die Ausgabe erfolgt in BAR, RPM und auf 250rpm umgerechnete BAR-Werten.
Nach dem Test MKL löschen und Exzenterwellensensor neu kalibrieren.

Ich habe meine modifizierte Variante des Programms hochgeladen und die originale von TwistedRotors

Hier das Changelog zu der bar Variante:

V2v1.2beta and V1v1.1beta

- these versions are beta, if you use them for testing, please provide feedback how they worked for you, they have not been tested in a real measurement yet.
- Implemented a function to measure baseline of the sensor and use it for calculation (before the value was derived by the assumption the sensor should provide 0,5V in idle as stated in it's description, it turned out the values is usually less than that, now the real sensor value is used
- Slight adjustment of the correction function
- Deleted some unnecessary functions and values

V2v1.1

- for V2 Tester, adjusted the correction function to fit Mazda values, more measurements required to improve results

v.1.0

- two versions: one for the old testers from 2018 (V1) and one for the newer version from 2020 (V2) as the new sensor seems to scale differently.
- added compensation to match compression values to the values of the OEM Mazda Tester
- reduced the measurements to 3, while only the middle one is regarded reliable, this gives a quicker output

v.04:

- complete rewrite of the code
- added function to output only the middle measurement of 5, as the rpm might not be stable in the first and last measurement, these values are not sent to output any longer
- deleted filter functions for unrealistic rpms
- removed the 000 Bar and RPM line in the initial screen

v0.2:

- added dead space correction for 13B MSP Renesis engine
- changed output from psi to bar with 1 decimal place
- added function to normalize values to 250rpm based on a second order polynomial based on the original Mazda diagram for rpm correction

-added some functions to filter non-realistic rpms (mainly for testing purposes, but as this shouldn't compromise speed I left it in) Image not found or type unknown