

Inkoppling och konfigurering av 0-5 Voltsignal till OBDX enheten

Typ av signaler som kan kopplas till 0-5 Volt ingången

0-5 Voltsignalen måste vara drivande vilket i praktiken innebär att den måste komma från en "intelligent" enhet som till exempel ett bredbandslambdainterface. Man kan alltså inte med framgång koppla in en signal som kommer direkt från en resistiv givare. Anledningen är att signaler som kommer direkt från resistiv givare (som dom flesta motormonterade givare är) inte är linjär och det finns ingen möjlighet att linjärisera signaler i appar som till exempel OBD Torque eller OBD Fusion.

Vart man kopplar in 0-5 Volt signalen

Man kan koppla in en 0-5 Voltsignal på den ingången där man normalt sett ansluter insugstemperaturgivaren. Vill man mäta både en 0-5 Voltsignal och insugstemperatur så kan man välja att koppla in insugstemperaturgivaren på ingången för omgivningstemperaturgivaren och då tappar man möjligheten att mäta omgivningstemperatur istället.

Förberedning av hårdvaran för att kunna mäta 0-5 Volt signal

För att mätning av 0-5 Voltssignalen ska fungera behöver man ta bort en "jumper" kallad JMP1 på kretskortet enligt bilen nedan:

Konfigurering av OBD Torque appen för att tolka 0-5 Volt signalen

För att visa rätt mätenhet i appen (och inte bara ADC steg) behöver man konfigurera upp en egendefinierad PID. Det finns en video på hemsidan (www.stannumembedded.com) som visar hur man lägger till en egendefinierad PID men för att kunna få fram rätt mätenhet i appen krävs djupare förklaring av formeln som ska skrivas in:

Som exempel används 0-5 Volt signalen som AEM:s bredbandslambdasondsenhet 30-0300 genererar. Tabellen visar hur man ska tolka ut AFR och lambda från spänningsnivån:

0-5V Analog Output Scaling Table		
Volts	Lambda	AFR (Gasoline)
<0.50	SENSOR NOT READY	
0.50	0.58	8.50
0.75	0.62	9.09
1.00	0.66	9.69
1.25	0.70	10.28
1.50	0.74	10.88
1.75	0.78	11.47
2.00	0.82	12.06
2.25	0.86	12.66
2.50	0.90	13.25
2.75	0.94	13.84
3.00	0.99	14.44
3.25	1.03	15.03
3.50	1.07	15.63
3.75	1.11	16.22
4.00	1.15	16.81
4.25	1.19	17.41
4.50	1.23	18.00
>4.50	SENSOR ERROR	

Table 1: Encoding av 0-5V signal från AEM 30-0300 (urklipp ur 30-0300 manualen)

Det man behöver räkna ut för att kunna visa AFR via OBD Torque är offset samt skala. Skalan räknas ut som hur många AFR det är per volt. AFR anges mellan 8.5 och 18.0 då utspänningen är mellan 0.5 och 4.5 Volt. Överföringsfunktionen blir så här:

$$Scale = \frac{[AFR]}{[V]} = \frac{18.0 - 8.5}{4.5 - 0.5} = \frac{9.5}{4.0} = 2.375 AFR/V$$

Offset:en är egentligen 8.5 AFR vid 0.5 Volt men eftersom OBD Torque (och OBD Fusion) helst vill börja räkna från 0 Volt får vi extrapolera offset:en ner till 0 volt. Vi räknar alltså ut vilket AFR 0 V borde motsvara (om den spänningsnivån hade varit giltig).

$$Offset = 8.5 - (2.375 \times 0.5) = 8.5 - 1.1875 = 7.3125 \text{ AFR}$$

Nu har vi allt vi behöver för att konfigurera OBD Torque att presentera AFR från en 0-5V signal från ett AEM bredbandslambdainterface:

$$OBD \text{ Torque Equation} = \frac{(Scale \times 5 \times (A \times 256 + B))}{1024} + Offset$$

För att interface exemplet AEM 30-0300 bredbandslambdainterfacet så startar man OBD Torque, går till: "Settings" => "Manage extraPID:s/Sensors" => "Add custom PID" och anger följande:

OBD2 Mode: 01ab
Long Name (used in menus): AirFuelRatio (endast ett förslag)
Short Name (used in displays): AFR (endast ett förslag)
Minimum value (typically 0): 0
Maximum value: 1024
Scale factor: x1
Unit type: AFR (endast ett förslag)
Equation: ((2.375*5*(A*256+B))/1024)+7.3125

Lämna resten av raderna **tomma** eller med default inställningar och avsluta med "OK"

Onödigt vetande

För er som inte gillar magic numbers och vill ha en förklaring på vad A, B, 5:an, 256:an och 1024:an betyder i OBD Torque ekvationen så kommer här en förklaring. Detta är ingen kunskap som behövs för att konfigurera upp OBDX systemet men ibland kan det vara kul med lite överkurs.

$$OBD \text{ Torque Equation} = \frac{(Scale \times 5 \times (A \times 256 + B))}{1024} + Offset$$

OBDX enheten har 10 bitars upplösning på AD omvandlaren, dvs. den periferienhet i mikroprocessorn som läser av de analoga insignalerna (givarvärdena) delar upp det avlästa spänningsvärdet i ett 10 bitar långt binärt tal.

För att räkna ut hur stort tal man kan beskriva med ett visst antal siffror med viss talbas så räknar man så här:

$$Största \ värde = Talbas^{Antal \ Siffror}$$

I vår decimala värld så känns det naturligt att man kan beskriva värden upp till 999 med tre siffror (egentligen kan man beskriva 1000 tillstånd men vi brukar inte räkna nollan till vardags). Vi kan räkna ut detta så här:

$$Största \ värde \ man \ kan \ beskriva \ med \ 3 \ decimala \ siffror = Talbas^{Antal \ Siffror} = 10^3 = 1000$$

På samma sätt kan vi räkna ut hur många tillstånd man kan beskriva med ett 10 siffror långt binärt tal. Ett binärt tal (en bit) kan bara anta värdet "0" eller "1" vilket är 2 tillstånd. Talbasen är alltså 2 och antal siffror i vårt fall är 10:

$$Största \ värde \ man \ kan \ beskriva \ med \ 10 \ binära \ siffror = 2^{10} = 1024$$

När OBDX enheten kommunicerar ut 0-5 Volt signalen via OBD kontakten så skickas värdet grupperat i bytes. En byte är 8 bitar lång och eftersom vi har en 10 bitars AD omvandlare så räcker inte en byte utan vi måste använda 2, **A** och **B**. Datat som ligger i byte A är de två bitar som ligger längst till vänster i det 10 bitar långa 0-5 Volt värdet och dom är värda $2^8 = 256$ gånger mer än dom bitar som ligger i byte **B** enligt samma logik som vi har när vi räknar med decimala tal; i talet 123 så är ju 1:an värd 10 gånger mer än 2:an som i sin tur är värd 10 gånger mer än 3:an. Därför måste vi multiplicera **A** med **256**.

Nu har vi fått en förklaring på alla delar i OBD Torque ekvationen. **A**, **B**, **5**, **256** och **1024**.