Acceleration Enrichment (AE)

"A quick shot of fuel"

GN and Turbo Regal **Dennis Leek** (dmdl@snip.net) 7-23-2005 Rev 2

The information contained in this article is based on the 1986/87 Buick turbo V6 ECM and the factory chip (labeled ACXA 0942). Memory addresses are shown as well as the formulas to calculate data values for those who are interested.

Introduction: When you push the gas pedal down quickly, the ECM enters a mode called Acceleration Enrichment (AE). The purpose of AE is to add fuel to match the sudden "in rush" of air. The ECM doesn't stay in this mode very long because it only needs to add enough fuel to compensate for the change (increase) in throttle position. Some refer to this function as a "pump shot" which is a term originating from carbureted engines. When the gas pedal is depressed about half way or more, the ECM switches to a fuel mode called Power Enrichment (PE), so it will add fuel for AE first, then go into PE.

The turbo V6 ECM can add fuel during acceleration via two different methods. The first is the additional fuel determined solely by the amount of **throttle (TPS) increase**. The ECM looks at the change in throttle position, determines how much fuel to add, then sends the additional fuel signal to the injector controller on a port *separate* from the normal injector pulse-width signal. The ECM checks the throttle position sensor (TPS) at a very fast rate and as long as the TPS% is increasing, the additional fuel is supplied. When the TPS% stops increasing, the TPS AE fuel is stopped.

The other method the ECM uses is based on **load (LV8) increase.** LV8 AE fuel is calculated as a percentage (of the normal injector pulse-width) to *add* to the otherwise normally calculated injector pw. The LV8 AE fuel is applied in full only long enough for each injector to apply it. The additional fuel is then reduced on each successive crank cycle until it reaches zero. *The LV8 portion of AE is disabled in the 85 and 86/87 factory programs.*

Acceleration Enrichment Due To TPS Increase:

[$3AB9] The TPS increase needed to enable AE mode is only 0.78% ($45F). When the throttle is pushed down, the ECM sets the AE flag ($07 bit 3) then calculates how much additional fuel to supply based on how far the gas pedal was pushed. The amount of TPS increase is multiplied by a table value ($5B2) with the result representing the amount of intended AE fuel.

AE TPS Delta Multiplier ($5B2)			
Coolant Temp (°F)	Data Value (hex / decimal)		TPS Delta Multiplier
-40	0A	10	0.039
3	09	09	0.035
46	08	08	0.031
90	07	07	0.027
133	06	06	0.023
176	05	05	0.020
219	04	04	0.016
262	03	03	0.012
305	03	03	0.012

AE TPS Delta Multiplier vs. Coolant Temp (0 – 99%)
– output not stored
Table provides ability to vary TPS AE fuel with coolant temperature.
Table entries are multipliers for TPS delta (amount of TPS increase).

Index = current coolant temperature

Table value formula: `AE_TPSΔMult = N/256`

During the time it takes to push the accelerator pedal down, the ECM can read the Throttle Position Sensor many times. Each time it is read, the TPS AE injector pulse-width size is calculated based on TPS increase since previous read. Each new pw size is added to the total of the previously calculated AE pulse-widths ($98) and applied as AE fuel. As long as the TPS% continues to increase, this process continues.

[$3AD9] After the ECM determines the TPS AE pw, it checks to see if it is above a *minimum* size. The minimum pulse-width is 0.49 msec ($43C). (The formula is: `InjPW = N/65.536.)` If the pw is below the minimum, it is changed to zero and no additional fuel is applied. This check is needed to shut off the AE fuel when it is being reduced because the TPS% no longer increases (described later).

The next thing the program does [$3AE5] is check if the fuel mode DFCO is also enabled. This is Deceleration Fuel Cut-Off and is applied when you *lift* your foot from the accelerator. An example of when the ECM would be in DFCO and AE at the same time is, you accelerate then let up, then accelerate again. In this case, the ECM would be in just AE, then just DFCO, then AE with DFCO. The program was designed to increase the AE injector pulse-width by some fixed amount to offset the DFCO fuel reduction, but this value ($447) in the 86/87 chip is 00.

Next, the program checks the LV8 value to see if the load has increased. This is for LV8 AE but to maintain continuity, let's hold off the details of LV8 AE for just a moment. After the ECM determines the amount of fuel to add because of TPS increase, it then performs its normal fuel calculation. The calculated injector pw (excluding anything for TPS AE) is sent to the injectors, then the program resumes TPS AE processing.

[$3FB8] The AE pulse-width is checked to see if it is below a *maximum* value. If it exceeds the max value, the calculated pw is replaced by the max pw value. Finally, the TPS AE pw is sent to the injector controller on a port separate from one used for the normal injector pw signal and more fuel is applied.

Maximum TPS AE Injector PW:

There are three tables and a single data item used to limit the size of the TPS AE pulse. One table contains maximum injector pulse-widths, and the other two tables and data item contain scalers (multipliers) used to adjust the max pw based on variable conditions.

Maximum TPS AE PW ($5BD)			
Coolant Temp (°F)	Data Value (hex / decimal)		Max PW (msec)
-40	70	112	1.709
-18	68	104	1.587
3	60	96	1.465
25	58	88	1.343
46	52	82	1.251
68	4F	79	1.205
90	4D	77	1.175
111	4B	75	1.144
133	4A	74	1.129
154	49	73	1.114
176	48	72	1.099
198	47	71	1.083
219	46	70	1.068
241	45	69	1.053
262	43	67	1.022
284	41	65	0.992
305	40	64	0.977

Maximum TPS AE PW (0 – 3.89 msec)
– output not stored
Table sets a limit on the maximum size of the TPS AE pulse width based on current coolant temperature.

Table output can be scaled based on current throttle position ($5CE) and/or engine runtime ($5DF).

Table value formula:
`MaxTPS_AE_PW = N/65.536`

This table establishes the base maximum TPS AE pulse-width size. The program was written so the max pw could be easily adjusted up or down, however the first scaler ($5BC) is a single data item with a value (multiplier) of 1. Next is a table of multipliers based on TPS position ($5CE) but all the table values are = 1.00. This table is shown on the next page. The final scaler for max pw is a table whose output varies with engine runtime ($5DF). Well, this table is disabled (output not used). This leaves the base pw value from the Max TPS AE table ($5BD) to be used as is.

TPS Scaler of Max TPS AE PW ($5CE)			
TPS%	**Value**		**Multiplier**
00	80	128	1.00
6.25	80	128	1.00
12.50	80	128	1.00
18.75	80	128	1.00
25	80	128	1.00
31.25	80	128	1.00
37.50	80	128	1.00
43.75	80	128	1.00
50	80	128	1.00
56.25	80	128	1.00
62.50	80	128	1.00
68.75	80	128	1.00
75	80	128	1.00
81.25	80	128	1.00
87.50	80	128	1.00
93.75	80	128	1.00
100	80	128	1.00

TPS Scaler of Maximum TPS AE Pulse-Width (0 – 1.99)
– output not stored

Table provides a plus or minus adjustment to the maximum AE injector pulse width based on current throttle position.

Table value formula:
`TPS_ScalerMaxTPS_AE_PW = N/128`

Runtime Scaler of Max TPS AE PW ($5DF)			
Runtime	**Value**		**Multiplier**
0 sec	93	147	1.15
8 sec	93	147	1.15
16 sec	8F	143	1.12
24 sec	8C	140	1.09
32 sec	88	136	1.06
40 sec	84	132	1.03
48 sec	84	132	1.03
56 sec	82	130	1.016
64 sec on up	80	128	1.00

Runtime Scaler of Maximum TPS AE PW (0 – 1.99)
– output stored in $96
Table provides a plus or minus adjustment to the max AE injector pulse width based on engine runtime (since it was last started). This table is only used when the coolant temp is below a certain temperature ($795).

Table value formula:
`Runtime_ScalerMaxTPS_AE_PW = N/128`

[$F2D6] The ROM portion of the program reads the Runtime Scaler table ($5DF) and stores the result ($96). [$38BC] Before the table value is used, the coolant temp is compared to a limit. The 86/87 factory program has max temp set to – 29 ºF ($795), effectively disabling the table. The ROM program still reads the table but the PROM program replaces the table's output value with a hex 80 (a multiplier of 1).

[$3FDF] The final step in producing the TPS AE fuel is to make what is labeled a "Battery Voltage Offset" adjustment. Since the engine is running, the alternator output determines this value so technically, it is the "Alternator Voltage" adjustment. In any case, five times each second, a table ($4C1) is indexed by the amount of voltage at the battery ($35) and the table output value is added to the TPS AE injector pw *twice.* Then the final number (pw) is sent to the fuel injector controller chip. (The Battery Offset table is shown on the next page.)

The theory behind this table is that during normal fuel delivery (not AE), it provides a method of fuel compensation should the voltage increase/decrease at the injectors. The problem is the same table is also used with TPS AE and it can cause the calculated AE pw to *double* in size. Look at the Max AE PW size for 176 ºF (table $5BD). It is 1.1ms. Normal battery voltage is between 12.8V and 14.4V, so a pw of about 0.945ms is added which makes the pw sent to the injectors equal 2msec (double the table value).

For the "chip tweakers": If you change the Battery Offset table values, keep in mind it will affect both TPS AE and the normal (non-AE) injector pulse widths. Changing to a different size injector usually requires changing this table. Also when making changes, don't forget that its added twice.

Injector Offset vs Battery Volts ($4C1)			
Volts	**Value**		**PW (msec)**
00	FF	255	7.77
1.6	FF	255	7.77
3.2	FF	255	7.77
4.8	FF	255	7.77
6.4	FF	255	7.77
8.0	6C	108	3.29
9.6	42	66	2.01
11.2	31	49	1.49
12.8	23	35	1.07
14.4	1B	27	0.82
16.0	15	21	0.64
17.6	0F	15	0.46
19.2	0F	15	0.46
20.8	0F	15	0.46
22.4	0F	15	0.46
24.0	0F	15	0.46
25.6	0F	15	0.46

Injector Offset vs. Battery Voltage (0 – 7.77 msec)
- output stored in $8C

Table provides compensation for changes in voltage at the fuel injectors. The table output is added to the normal injector pulse width and again to the TPS AE injector pw.

The pulse-width size shown here is the pw added after the table value is added *twice*.

Table value formula:
`InjOffsetBattery = N/32.768`

TPS AE Example:

While driving, you accelerate to pass a slower car. Let's say its a Mustang. Now this is relevant because we all know it doesn't take much to pass a Mustang. I'll say that just a 30% increase in throttle is needed. The first change in TPS that the ECM detects is just a portion of the total accelerator pedal travel. Let's use 4.3%. The ECM uses binary numbers but decimal numbers make more sense to us, so that's what I'll use. The 4.3% change in TPS is represented by the number 11_{10} (4.3 x 2.56). With the coolant temp at 176 °F, the AE TPS Delta Multiplier (from table $5B2) is 05. The two numbers are multiplied (11 x 5 = 55) and the result (55_{10}) represents the amount of AE fuel to add. This equates to an injector pulse-width of 0.839 msec (pw = N/65.536). This is below the maximum for this temperature, so the first "shot" of AE fuel is for 0.00839 seconds. Not a very long time.

On the next pass through the program, the ECM detects say, a 9.4% increase in TPS over the last reading. (For the purpose of AE, the ECM is monitoring the *change* in TPS, not the total travel.) The TPS delta of 9.4% is represented by 24_{10} (9.4 x 2.56). When multiplied with the TPS Delta table value (05), the product is 120_{10}. This is added to the previous TPS AE pw, resulting in the number 175_{10} (120 + 55). Converting 175 to injector pw yields 2.67 msec. This new pw *does* exceed the maximum so it is changed to 1.099 msec (72_{10}), the max pw for this temperature as determined by table $5BD.

As each additional TPS increase is read, the program adds to the previously determined AE pw, but since it already exceeds the max size, the actual AE fuel applied from this point until the TPS stops increasing, will be the max pulse-width and not the calculated pulse-width.

A typical acceleration has the ECM in AE for about 1/3 rd of a second. After that, the ECM usually enters its Power Enrichment (PE) mode and the PE tables control the fueling until you let up on the gas pedal.

Observation: As shown in the example, it doesn't take much TPS increase before the max AE pulse-width is reached. To show this, I charted TPS increases of 2% - 7% for the 90 – 219 ºF range and compared the calculated pw to the factory values for max pw. The following table shows the result.

TPS Delta	90º	133º	176º	219º	Coolant Temp
02%	0.547	0.469	0.391	0.313	TPS AE PW (msec)
03%	0.820	0.703	0.586	0.469	
04%	1.094	0.938	0.781	0.625	
05%	1.367	1.172	0.977	0.781	
06%	1.641	1.406	1.172	0.938	
07%	1.914	1.641	1.367	1.094	
Max PW =	1.175	1.129	1.099	1.068	

AE PW for TPS Delta vs Coolant Temp
The actual max TPS increase before exceeding the limit at 176 ºF is 5.63%. As you look across the chart, you see that at any of these temperatures a TPS increase above 5 - 6% exceeds the max pw (shown in red). A typical acceleration has the TPS increase 40% or more, so the TPS AE fuel for most of the acceleration will be the maximum pulse width and *not* the calculated pulse width.

What happens when the TPS stops increasing?

The program has provision for either turning off the AE fuel or just reducing it when the TPS% stops increasing. At this point, the TPS% either stays the same, goes down a little, or more than a little. The TPS decreasing "a little" is defined as less than the amount needed to turn off AE, which is 1.172% ($462). This doesn't sound like very much, but keep in mind, the ecm is reading the TPS at a very fast rate (every 12.5msec). Typically, when you stop pushing down the accelerator the ecm "sees" the TPS% initially stay the same, then go down slightly.

When the "less than turn-off" condition is detected, a table ($5A8) of AE PW reduction coefficients is read. The coefficient in this case is the opposite of a multiplier. By that I mean, the table output represents the percentage of the TPS AE pw reduction. All the table entries have a coefficient of 0.688 which reduces the AE pw to 31.2% of its previous value.

AE TPS PW Decay Rate ($5A8)			
Coolant Temp (°F)	Data Value (hex / decimal)		TPS Delta Coefficient
-40	B0	176	0.688
3	B0	176	0.688
46	B0	176	0.688
90	B0	176	0.688
133	B0	176	0.688
176	B0	176	0.688
219	B0	176	0.688
262	B0	176	0.688
305	B0	176	0.688

AE TPS Pulse-Width Decay Rate (0 – .996)
- output stored in $9A

Table provides TPS AE pulse width reduction due to a TPS decrease of less than 1.172% ($462).

Index = coolant temp

Table value formula: `DecayRate= N/256`

After each reduction, the pw is compared to the minimum size of 0.49msec ($43C) and if below, it is changed to 00. When the ecm detects the TPS has stayed the same for several cycles or it went down 1.172% ($462) or more, it shuts off the TPS AE fuel completely.

LV8 Delta AE:

For the purpose of AE, LV8 Delta is not just a change in LV8 but is defined as the difference between the calculated LV8 and the *filtered* LV8 values.

[$39FC] LV8 is a number the ECM uses to represent the "load" on the engine. It is derived by multiplying RPM and airflow. The airflow is calculated by multiplying the current MAF Table value by the current MAF Table Scaler. The LV8 value is then scaled with a reference value of 1.04 ($435). [$EC09] To more smoothly track the load changes, the program filters the calculated LV8. The filtered value is a sample or percentage of the LV8 that is taken at a rate slower than that used to calculate the current LV8. The current LV8 is calculated every 12.5ms. Every 0.3 seconds ($45B), 50% ($45C) of the current LV8 is used as a sample and the filtered version is updated. This process smoothes the changes and eliminates the affect of short duration changes. LV8 delta used for AE is the current LV8 minus the filtered LV8.

[$37A6] An LV8 delta of greater than 48_{10} ($45D) will enable the LV8 AE controls. When this load increase is detected, a table of "LV8 Factors" is read using the LV8 delta above the minimum as the index. The output from this table ($57F) represents a percentage of the normal injector pulse-width. The LV8 Factor percentage is multiplied by the output from another table controlled by coolant temperature ($585). The result ($95) is doubled then used as a multiplier for the otherwise normally calculated injector pw. *The LV8 Factor table in the factory chip contains 0's, so no LV8AE fuel is added.*

The table indexes shown here are both LV8 delta and the LV8 delta above minimum values.

LV8 AE Factor vs. LV8 Delta Above Min ($57F)

LV8 Delta	LV8 > Min	(hex)	(dec)	Factor
48 - 111	00 - 63	00	00	0.00
112 - 175	64 - 127	00	00	0.00
176 - 239	128 - 191	00	00	0.00
240 - 255	192 - 255	00	00	0.00
256	256	00	00	0.00

LV8 AE Factor vs. LV8 Delta Above Min (0 – 100%) – output not stored
Table produces a number relative to the increase in load. This table's output is multiplied by the output of the next table ($585).
Index = LV8 delta *above the minimum* ($45D)
Table value formula: `AE_LV8Factor= N/256`

The final LV8 AE value represents a percentage of increase to the injector pw. [$3F1F] The otherwise (non-AE) calculated injector pw ($83) is multiplied by the LV8 AE value and the result is then added to the injector pw so that it is increased by the LV8 AE percentage.

LV8 AE Factor Multiplier vs. Coolant ($585)

Coolant Temp (°F)	Data Value (hex / dec)		Multiplier
-40	00	00	0.000
3	00	00	0.000
46	00	00	0.000
90	05	05	0.039
133	05	05	0.039
176	05	05	0.039 (4%)
219	05	05	0.039
262	05	05	0.039
305	05	05	0.039

LV8 AE Factor Multiplier vs. Coolant temp (0 – 200%)
– output stored in $A4

Provides ability to vary the LV8 AE fuel with coolant temperature.

Table value formula:
`LV8AE_FactorMult = N/128`

When LV8 AE fuel is applied, the increased injector pulse width is used only long enough for all the injectors to apply it, then it is reduced by the output from the next table.

LV8 AE Fuel Decay Rate vs. Coolant ($58F)

Coolant Temp (°F)	Value (hex / dec)		% Reduction
-40	FF	255	100
3	FF	255	100
46	FF	255	100
90	FF	255	100
133	FF	255	100
176	FF	255	100
219	FF	255	100
262	FF	255	100
305	FF	255	100

LV8 AE Fuel Decay Rate vs. Coolant Temp (0 – 100%)
- stored in $9B
Determines how much to reduce the LV8 AE fuel on each crankshaft cycle after the AE fuel was applied.

Table value formula:
`LV8AE_DecayRate = N/256`

Tip for the "chip hackers":
If you want to enable LV8 AE fuel, keep in mind the final LV8 AE value is a percentage of the normal injector pw. As such, the smallest value to cause an increase is 01. This means the product of multiplying the two LV8 AE table values must equal at least 256_{10} (0100_{16}). As an example, to use the factory LV8 AE Multiplier table ($585) values, the LV8 AE Factor table ($57F) value must be at least 26_{10} ($1A_{16}$). [5 x 26 = 130][130 x 2 = 260][260/256=1.015] (the fraction is discarded)
`LV8AE=((LV8_Mult x LV8_Factor)x2)/256` ***Result must = 1 or more to increase fuel***