

# HP34401 Notes

The HP34401A bench DVM is my favourite among the family of 6.5 digit DVMs. This naturally makes me curious about how it works.

The HP service manual is helpful to a point but not the quality of *classic* HP service manuals, plus it doesn't always seem right and there is a big caveat. Everything here was determined by reverse engineering. There may (will) be mistakes - I'm writing as if all is fact, just because too many "I believe that" gets annoying.

I was primarily interested in how the ADC worked on the DC volts ranges. The various measurement types are split into three classes "DC-like", "AC-like" and "time-like" (classed as 0 to 2 respectively).

The primary hardware is an integrator U402-A. The inputs are summed from (a) signal (which can be switched to ground (MZ), input amp (MC) or pre-charge DAC amplifier (PRE)) or (b) positive fixed current (from a precision voltage source through a precision resistor) (c) negative fixed current. The integrator output feeds to the channel 0 AD input on the 80C196KB. It also feeds indirectly to the ASIC. In operation at all times either the positive (b) or negative current is turned on (c) (this is not made clear, but implied from the calculation in the rundown gain calibration test (error 605). The ASIC checks the output every 2.66us and switches between positive and negative current inputs as required to drive integrator output toward 2.5V (input range is 0-5V).

While one sign (presumed +) of current is enabled, the ASIC counter counts pulses (assumed to be the clock) in a 256 state internal counter. Pulses are gated to not be counted while the opposite sign is active. Overflow from this 256 state counter feeds P2.3 (T2CLK) on the CPU (and because they are at least divided by 256 they fall within the timing requirements of this counter). The CPU for DC measurement is configured to interrupt on T2-capture, but not on internal AD done. By raising a sync pulse (positive edge) on P2.4 of the CPU, T2Capture latches the state of the T2 clock in CPU and also an internal U501 latch captures the value of the 256 state counter in the ASIC into a register mapped to RAM at 1FFA (byte). The values of both counters can be concatenated bitwise to give a complete 24-bit count. The counter is NOT reset by this operation, the current value is copied to a separate latch (manual suggests otherwise but this wouldn't work for the long period integrations e.g. 100NPLC).

A sampling process is as follows: Two Timer1 counts [1] before the sync pulse to latch T2 and the ASIC count, the AD in the 80C196KB triggers its AD sample and hold selecting channel 0 (output of the integrator). Because the CPU has clock of 12MHz is "fast" (err manual says fast is >6MHz) the internal AD uses a clock pre-scaler and its sample and hold time would appear to be 15 state times, with 1 state time to start the process. Thus the S+H gate closes at the same time as the count is latched. The sync also triggers a T2 count capture and subsequently a T2 capture interrupt. The handler for this interrupt assembles the complete count data and collects the internal AD result (interestingly no attempt is made to check that the conversion is complete e.g. from AD status, or AD done interrupt). Judging by a pointless left shift 15 places of 0 instruction (which is slow) the code execution path has been made sufficiently long that this is a given (although I have not been able to verify this, partly due to lack of clear data).

All measurements are differential and process is as follows:

- Settling time (200us) after selecting a new input switch
- AD start followed by sync pulse is used to record initial values
- CPU HSO timer used to implement fixed delay e.g. 15000 T1 ticks for 1PLC at 50Hz
- start AD after fixed delay
- 2 T1 ticks after AD raise SYNC

HSO control scheduled this way is accurate +/-50ns. Long times > 10PLC are split up (and increments over each period summed) to avoid risks from complete 24-bit counter wrap arounds. In auto-zero mode for these long periods it appears that instead of a block of MZ and a block of MC there are alternating blocks.

Interim results are computed in long integrations (possibly to allow ranging decisions to be made more quickly?) so the actual structure is two nested measurement loops up to 10x 10x in NPLC 100 case, but the

finished flag is only set when the full period has elapsed. For reference with a 50Hz supply

NPLC	T1-cycles(50Hz)	T1-cycles(60Hz)	Inner Loop	Outer Loop (wrap avoidances and MC/MZ switch)
0.02	300	300	1	1
0.2	2250	2250	1	1
1	15000	12500	1	1
10	15000	12500	10	1
100	15000	12500	10	10

The astute observer will see the T1-counts for < 1PLC are the same for both line frequencies (and the first is correct at 50Hz and the second at 60Hz). I cannot explain this other than by assuming it was decided that for < 1 full cycle the precise timing didn't matter. The longer periods which are integral multiples of the line period are exact.

To subtract two results we need to know about rundown gain. This may be standard but I don't have a reference to hand:

With input voltage  $v$ , if we integrate for a time  $t$  clocks, and the count at start is  $c_0$  and end is  $c_t$  then voltage at output of integrator  $V_t$  will be given by

$$V_t - V_0 = v * a * t + b * (c_t - c_0) - b * (t - (c_t - c_0))$$

$a$  is proportionality constant for measurement current & integrator capacitor  $b$  is proportionality constant for the fixed voltage and resistor & integrator capacitor used in the multislope converter.

So if  $V = V_t - V_0$  is difference in integrator voltage over period and  $C = c_t - c_0$  is count diff

Then

$$V = (a * v - b) * t + 2 * b * C$$

Or

$$(a * v - b) * t = 2 * b * C - V = (\text{rundown gain}) * C - V$$

If we now apply  $v=0$  (call this MZ for measure zero, in contrast to MC for measure client) and repeat

$$-b * t = (\text{rundown gain}) * C_{MZ} - V_{MZ}$$

So subtracting the two

$$a * v * t = ((\text{rundown gain}) * C_{MC} - V_{MC}) - ((\text{rundown gain}) * C_{MZ} - V_{MZ})$$

From which  $v$  may be derived.

Depending on the Auto-Zero setting one MZ measurement is made at the start of a block of measurements or MC and MZ alternating. For long > 10PLC measurements the alternating occurs within the measurement period after blocks of 10PLC. After disconnecting MZ, pre-charge is applied for a fixed time (busy loop) and then the pre-charge switch opened before connecting MC (and thus the input amplifier).

Rundown gain may be estimated by selecting MZ and integrating over two different time periods  $T_1$  and  $T_2$  obtaining internal 80C196KB AD count differences  $V_1$  and  $V_2$  and multislope count differences  $C_1$  and  $C_2$

$$\text{rundown gain} = (V_1 * T_2 - V_2 * T_1) / (T_1 * C_1 - T_2 * C_2)$$

This is used to test the AD functionality, but not as part of the calibration, the NVRAM stored value is used to reduce measurements. Invalid results (e.g. same for both times) give errors 605/6. The calibration is repeated 8 times and the similarity of results checked (error 607) to ensure the difference between maximum and minimum values obtained is less than 0x100 otherwise the test fails.

For the purpose of reducing measurements the rundown gain is used as follows. It should be noted that fixed precision arithmetic is used throughout the result reduction process.

$$\text{Value diff} = (\text{count diff}) * (\text{rundown gain param } \#70) / 4 - (\text{Ad count diff})$$

The divide by 4 is simply to allow the gain to be expressed in a finer increment of 0.25 rather than 1.0.

## Practical points

You can see the gain with a SCPI peek

diag:peek? -2,70,0 (mine is 1694)

and after a measurement (e.g. a READ?) you can see the value diff

diag:peek? -7,0,0

Next stage is to turn this raw count into an expression voltage at the AD input. This is done by first an affine transformation ( $y=ax+b$ ) in fixed precision arithmetic. This is expressed as

$y = \text{round}((x - \text{offset}) * \text{mult} * 2^{\text{shift}} / 2^{32})$

where  $0 \leq \text{mult} < 2^{32}$ , and as usual in this form of representation to represent a factor  $c$ , the shift is chosen to be the smallest integer such that  $2^{\text{shift}} > c$  which is corrected by the  $\text{mult}/2^{32}$ . Two offsets are stored on each range, for front and rear terminals.

The coeffs from NVRAM must be adjusted for

- integration time (and line period if NPLC)
- to compensate for the non-linear adjustment at 10V input to ADC

Having thus determined a first value for the input to the ADC the non-linear correction may be applied. This has two terms one quadratic and the other cubic (effectively the cubic is a mid range correction). The quadratic coefficient is determined in the -10V calibration (so it is effectively the +ve/-ve voltage symmetry adjust). Because it is a quadratic correction the same adjustment direction will be made at +10V as at -10V so if coeff is positive this will increase the reading at +10V away from 0 and decrease the reading at -10V (toward zero). Thus the calibration has to include a compensatory step to ensure the +10V FS calibration is unchanged (the scaling factor is adjusted). The coefficient for the cubic is a mid range non-linearity correction is calibrated as part of the 500V DC cal step. The graph shows the form.

The input to the adc is expressed as an integer value which is multiplied by  $1e-7$  to give ADC input in volts. The actual meter inputs may require further multiplication by powers of 10 to correct for input amp gain, or divider ratio. Let the true value in volts be  $x$ . The corrections are implemented in fixed point arithmetic but for the sake of comprehension we use floating point expressions:

First stage quadratic correction (in counts) =  $0.10077 * \text{nlc1} * x^2$

Second stage correction (in counts) =  $\text{nlc2} * x * (2.691209 - x^2 * 0.02712)$

The two corrections are added to the ADC input count and this is the result. These fp values are approximate - because these adjustments have a granularity limited to 1 tick there seems little point in writing them out with loads of huge numbers.

It will be noted that only the first correction is significant at full scale, where with an input  $x=10V$ , the output reads  $10^8 + 0.10077 * \text{nlc1} * 100 = 1e8 + 10 + 10 * \text{nlc1}$ . Thus the raw multiplier from calib is corrected in presence of non-linear correction

$\text{mult} = (\text{calib mult}) * (\text{plc correction}) * 1e8 / (1e8 + \text{nlc1} * 10)$

This ensures that the same calib coeffs can be used irrespective of the non-linear correction choices.

## NVRAM Data format

The NVRAM storage is a packed bit stream (if a quantity requires 3 bits to store then this is exactly what the next 3 bits are used for). There are no headers so it is hard to read by hand as one need to know the exact bit-offset of the data of interest. Fortunately the firmware can extract items for us.

The range cal data can be extracted using diag:peek? -2, set, 0 Set numbers from lowest to highest range are

DCV 75-79

DCI 82-85

OHM 87-92

OHM4 94-99

ACV 104-108

Looking in more detail at one row, the DC 10V range is 77. On my meter (and obviously not yours as the calibration is probably different!) this is 2271461829,1,0,69,77,

The values are: multiplier, shift left count, 0, offset front terminals, offset rear terminals. I have no idea if the 0 can be given a value, but when in use in the meter this value is replaced by NLC2 (the second global nonlinear AD correction)

These values are applicable for the longest standard integration time. They are converted via factors for NPLC and line frequency (again note the 0.02 and 0.2 NPLC have the same factors for 50 & 60Hz), I suspect this is actually wrong, but since it is < 1 cycle it doesn't really matter).

We have an affine form  $\text{mult} * (x - \text{offset})$  applicable where  $x$  is measured over the longest integration time. If we integrate  $x$  over a fraction  $r < 1$  of the calibrated time we expect

ADC input count =  $\text{mult} * (x / r - \text{offset}) = (\text{mult}/r) * (x - \text{offset} * r)$

This is the NPLC scaling.

$\text{mult} \rightarrow \text{mult}/r$

$\text{offset} \rightarrow \text{offset} * r$

Since  $r < 1$  this requires no shift and can be expressed as  $(\text{offset multiplier})/2^{32}$ . The multiplier by  $1/r$  requires a  $2^k$  term since it is greater than unity. For minimal loss of precision it is desirable to ensure that the multiplier (unsigned) has its leading 1 as close as possible to the MSB. This may be achieved by

The scaling is thus

scaled multiplier =  $\text{mult}/r = 2^{(\text{shift})} * \text{mult} * 2^{\{\text{adjshift}\}} * \text{adjmult} / 2^{32}$

scaled offset =  $\text{offset} * r = \text{offset} * \text{adjoffset} / 2^{32}$

## NPLC scaling for 50Hz

NPLC	adjoffset	adjmult	adjshoft
0.02	0x000d1b71	0xa0000000	0x3
0.2	0x00624dd3	0xa6aaaaac	0xa
1	0x028f5c29	0xc8000000	0x7
10	0x1999999a	0xa0000000	0x4
100	0xffffffff	0x80000000	0x1

## NPLC scaling for 60Hz

NPLC	adjoffset	adjmult	adjshift
0.02	0x000d1b71	0xa0000000	0x3
0.2	0x00624dd3	0xa6aaaaac	0xa
1	0x02222222	0xf0000000	0x7
10	0x15555555	0xc0000000	0x4
100	0xd5555555	0x9999999a	0x1

An example might help here. Suppose we wish to use the 10V DC range to measure the voltage of a 1.5V cell using the front panel using a 10NPLC sampling window. The cal data is 2271461829,1,0,69,77, and assume a 50Hz line supply.

The 10NPLC adjust is

NPLC 10 adjoffset 1999999a, adjmult a0000000 adjshift 4

New offset =  $\text{round}((1999999a * 69) / 2^{32}) = \text{round}(0.1 * 69) = 7$

New shift =  $4 + 1 = 5$  (subject to adjustment)

New multiplier =  $2271461829 * a0000000 / 2^{32} = 1,419,663,643.15$

The binary representation of this begins 01... so to avoid precision loss we multiply by 2 and drop by one the shift. So

shift=4  
multiplier = 2,839,327,286

[So basically offset is divided by 10 and multiplier multiplied by 10 to account for the shorter integration time]

Now this meter has NLC1=27 and NLC2=4 (decimal) and as a result for a full scale AD input of +/-10V the first nonlinear correction results in a count value of  $10^8 = 10^8 + \text{NLC1} * 10$ . It should read  $10^7 * 10 = 10^8$  so the multiplier correction required is  $1e8/100,000,270$

Offset = 7  
Multiplier =  $2839,327,286 * 1e8/100,000,270 = 2,839,319,619$   
Shift =4

In the 7-5-2 firmware the coeffs used can be read out at location 15B2 and these are indeed the ones used. Let us now reduce a reading; suppose we measure a 1.5V cell and get a raw count of 1510615 (peek -7,0,0 value).

Reading =  $\text{round}(2^{(\text{shift})} * \text{mult} * (\text{count} - \text{offset}) / 2^{32})$   
=  $\text{round}(2^4 * 2,839,319,619 * (1510615 - 7) / 2^{32})$   
= 15,978,139

Now we must apply the non linear corrections; this suggests  $x = 1.5978139$   
first correction =  $\text{round}(0.10077 * \text{nlc1} * x^2) = \text{round}(0.10077 * 27 * (1.5978139)^2) = 7$   
second correction =  $\text{int}(\text{nlc2} * x * (2.691209 - x^2 * 0.02712))$   
=  $\text{round}(4 * (1.5978139) * (2.691209 - (1.5978139)^2 * 0.02712)) = \text{int}(16.75) = 16$

Result =  $15,978,139 + 7 + 16 = 15,978,162$   
So voltage = 1.5978162V

[1] A T1 count is 8 state times. One state time is two clock cycles (at 12MHz)