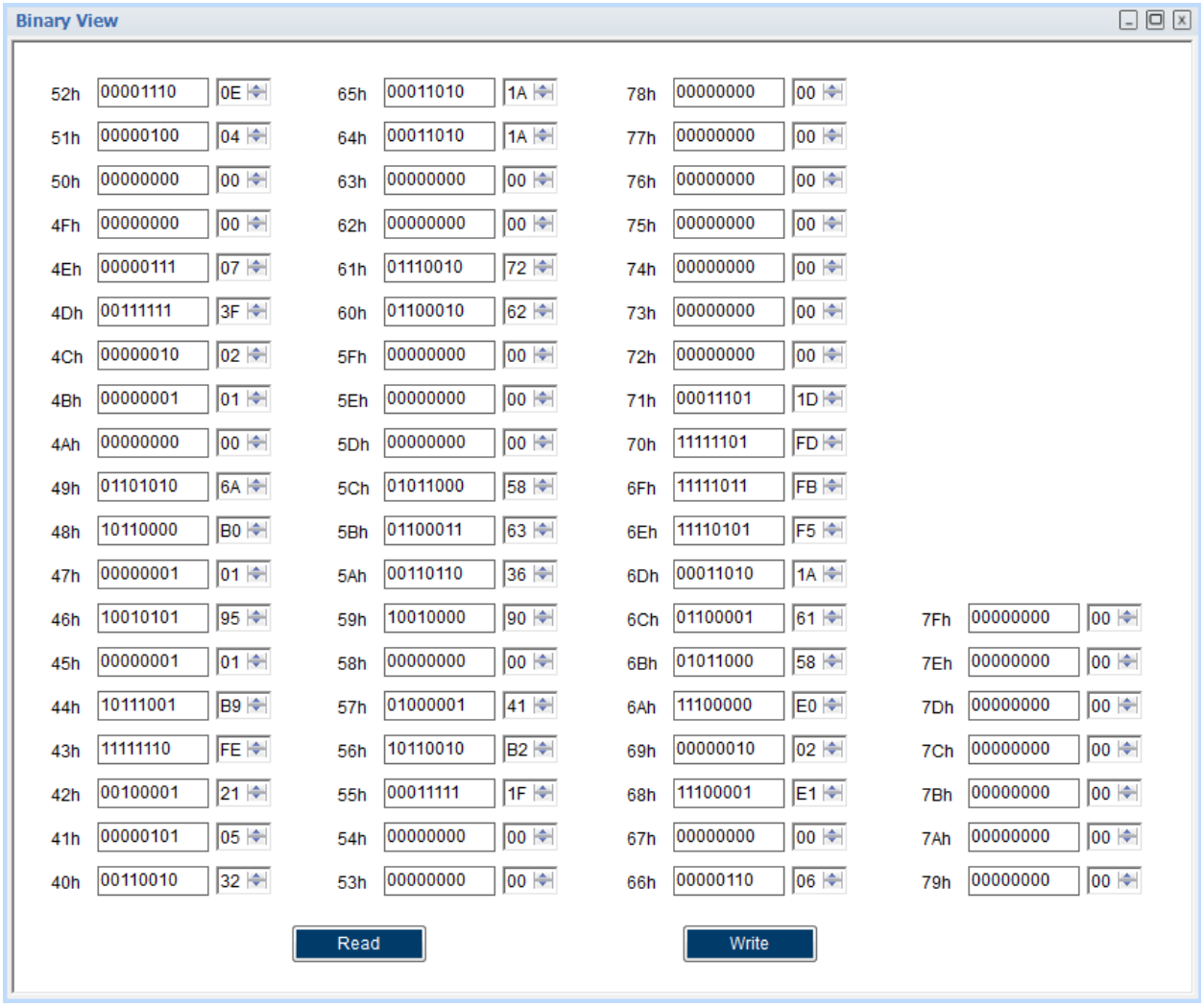
BMM150 memory view sample (from DD 2.0 GUI SW):



Reading trimmings:

dig_x1 = 0x00 = 0
dig_y1 = 0x00 = 0
dig_x2 = 0x1A = 26
dig_y2 = 0x1A = 26
dig_xy1 = 0x1D = 29
dig_xy2 = 0xFD = -3
dig_z1 = 0x58E0 = 22752
dig_z2 = 0x02E1 = 737
dig_z3 = 0xFBF5 = -1035
dig_z4 = 0x0000 = 0
dig_xyz1 = 0x1A61 = 6753

Reading raw data (uncompensated):

Hx = 0xFFC4 = -60 LSB
Hy = 0x37 = 55 LSB
Hz = 0xCA = 202 LSB
RHall = 0x1AAC = 6828 LSB

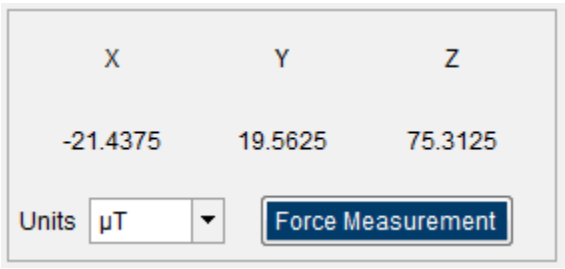
Manual computation of temperature compensated data:

Bx = bmm150_compensate_X_float(-60, 68280) = **-21.77 µT**
Formula output: `Bx = 1/16 ((-60 (((-3 ((6753×16384.0/6828-16384.0)×(6753×16384.0/6828-16384.0)/(2.684354560×10^8))+(6753×16384.0/6828-16384.0)×29/16384.0)+256.) (26+160.)))/8192.0+0×8.)`

By = bmm150_compensate_Y_float(55, 68280) = **-19.96 µT**
Formula output: `By = ((55 (((26 ((6753×16384.0/6828-16384.0)×(6753×16384.0/6828-16384.0)/(2.684354560×10^8))+(6753×16384.0/6828-16384.0)×29/16384.0)+256.) (26+160.)))/8192.0+0×8.)/16.`

Bz = bmm150_compensate_Z_float(202, 68280) = **75.74 µT**
Formula output: `Bz = (((202-0)×131072.0-(-1035×(6828-6753)))/((737+22752×6828/32768.0)×4.))/16.`

DD 2.0 GUI SW compensated data output using the API for the same sample:



Precision in the results is due to the floating-point computation in your application versus integer computation in DD 2.0 GUI SW.