Addendum

to the

Advanced Programming Guide

8. REAL

The REAL format selected for use within the Modula-2 System for Z80 CP/M is the so called hidden bit format. This format looks like:

```
byte:          0                 1                 2                 3
bit:   7              0 15              8 23             16 31             24
      +----------------+----------------+----------------+----------------+
      | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
      +----------------+----------------+----------------+----------------+
       ^ ^             ^                                                  ^
       | |<- exp(Z)  ->|                                                  |
       |               |<----            mant(Z)                     ---->|
       sign(Z)
```

sign(Z) = sign of the mantissa of the number.
exp(Z) = exponent to the base 2 of the number, biased by 80H. 0 value reserved for number 0.
mant(Z) = mantissa of the number with the hi bit always '1' and replaced by the exponent's low bit.
Z = sign(Z) * mant(Z) * $2^{(exp(Z)-80H)}$

In words, this means:

- A hidden bit coded floating point number consists of three parts: the **sign** of the mantissa, the biased **exponent**, and the **mantissa** without the hidden bit. Mantissa and exponent are both coded to the base 2.

- The sign bit is '1' for a negative mantissa, and therefore negative number.

- The exponent may be in the range -127 up to 127. This range is coded as 1..255. The number 0.0 is represented by a zero exponent because there is no way to represent it otherwise. The base (radix) of the exponent is 2, i.e. the mantissa value gets multiplied by $2^{(exponent-80H)}$. The exponent uses 8 bits. The low bit is stored in the hi bit of the second byte of the number.

- The mantissa is 24 bits wide. Because the hi bit is always '1', it is 'hidden', i.e. overwritten by the exponent's low bit. Its value is always in the range 0.5 <= mant(Z) < 1.0.

This format allows positive REAL numbers in the range $0.5 * 2^{-127} = 2^{-128}$ up to nearly $1.0 * 2^{127}$. In scientific notation, this corresponds to about 2.9387E-39 up to 1.70141E38. The resolution is 1 part out of 2^{24}, or about 7.2 decimal digits ($\log10(2^{24})$).

Besides the relatively large range for a four byte representation, the hidden bit format also offers the advantage that the significance of the digits sinks monotone, or formulated easier: To compare two hidden bit numbers, you have to care about the sign bit, the rest can be compared by dumb byte-by-byte comparisons. No knowledge of the rest of the representation and also no calculation (i.e. subtraction of the two numbers) is needed¢