**Universidad Autónoma de Guadalajara**

Ingeniería Electrónica Biomédica

System designing with Microprocessors

*“*Practice 6: Floats”

Jesús Arnoldo Zerecero Núñez

2885993

Practice 6. Floats

**Introduction**

Float type numbers are used in c coding to represent fractional or very large numbers. Each compiler stores float type numbers a different way, one which cannot be easily be read by human users. The IEEE issued a standard (754) to store, work with and translate float type numbers.

Many microcontrollers are unable to used float type numbers because of the bit size they represent (32 bits single precision, 64 bits double precision) and the difficulty of working with them.

In this practice, different float numbers functions were created, in order to facilitate their usage without utilizing standard libraries. These functions were created based on the IEEE 754 standard for floating point arithmetic:

* Integer to float conversion.
* Float to integer conversion.
* Addition of 2 float numbers.
* Subtraction of 2 float numbers.
* Comparison of 2 float numbers.
* Division of 2 float numbers.
* Module of 2 float numbers.

These functions work with float number no bigger than 32000 (16 bits). Trying to pass parameters bigger than 16 bits will result in the functions returning INFINITY or –INFINITY.

**Conclusion:**

Using floating point numbers has the advantage of dynamic range and precision. In applications were every nanosecond counts, storage precision is very important. In applications were huge numbers are used, it’s best if every digit is precise. For example, for medical applications were certain dose of milliliters must be injected into a patient, there is no room for error. Several processors won’t work floating point numbers, but will do their best to convert their fixed point numbers into floating point notation. However, this is not recommended as it takes a significant amount of cycles to accomplish and the result may involve the loss of data.