1) Inject 0x1FFFFFFF for the “counter” value in the variable window, then step thru the program only once to increment “counter”.

a) The adding the variable 0x1FFFFFFF, the value “counter” is now equal to 536870911

b) The value of the “counter” in the “Register” window is 0x20000000 which is set to R1

c) There is nothing was set for the APSR registers.

2) If your write all Fs (0XFFFFFFFF) in the Register value for “counter” then step thru the program once to increment “counter”

a) After changing the value of counter in the register, the value of counter in the “Locals” window is now equal to zero.

b) The flags set for APSR are both “Z”(triggers a zero condition flag) and “C”(triggers a carryover condition). Both are set to “1”

3) Change the “counter” variable type in your code to “unsigned”. Inject the values “0x1FFFFFFF” then step thru the program to increment the “counter” once:

1. The value of counter is now 536870912 after changing the “counter” variable to unsigned and stepping through it once.
2. No APSR flags are set in the registers

4) Change the “counter” variable type in your code to “unsigned”. Inject the values “0xFFFFFFFF” then step thru the program to increment the “counter” once:

1. The value of “counter” is 0 after stepping through once and setting the value to “0xFFFFFFFF”
2. The flags that are set for the APSR are “Z”(triggers a zero bit condition) and “C”(triggers a carryover condition). This factors in the overflow of the counter variable being incremented past the eight digitals, hence the “Z” and “C” flags.

5) Move the “counter’ variable outside of main (at the top of the file):

1. The scope of the counter variable is now a global one instead of a local one which was previously in the main function.
2. No, there is nothing displayed in the “local” window
3. We can view the counter variable in the “auto” window now.
4. From the “Auto” window, the “counter” variable address is at 0x20000000.

6) Change the source code to the following, then run the program still in the simulator:

a) The value of “counter” is 4.

b)We declare the pointer variable which we point to the “counter” variable address. From that, we can deference the point and increment systemically.

7) Change the setting of IAR to run the same program on the evaluation board:

a) The address of the counter variable is 0x20000000 since this is declared in main function.

b) The “Counter” variable is stored in RAM because this maintains values called for function local objects within the code.

C) The value of the “counter” variable is 4.