Cannot set user data location 10, always a very small number

Cannot set analog PWM output

**Issues**

void main()

{

printf("Console Test\n");

for (i=0; i<4; i++)

{

printf("Line %d\n", i);

}

printf("\n");

}

printf("\n"); Produces line feed in KMotion Console but not when passed through MessageReceived event

**Questions**

**Information**

There are several types of stopping that KMotionCNC supports:  
  
#1 "Stop" - this is like an emergency stop where the Interpreter immediately exits and all axes are disabled and all User C Programs are killed (except Thread#1).  The code refers to this as Abort.  Everything is left in an indeterminate state so KFLOP needs to be re-initialized to re-enable the axes  and the Interpreter should be restarted (with the Restart option in the call to Interpret) to re-initialize.  See:  
  
void CKMotionCNCDlg::OnEmergencyStop()   
  
#2 "Halt" - The intent here is to bring the motion to an immediate but controlled stop.  Internally a Feedhold is issued and the libraries wait until things come to a full stop.   The Libraries then re-wind the interpreter back to the exact point the machine actually stopped and the Interpreter is left in a resume-able state even though it may be in the middle of a line of GCode.  The stopped location and state is recorded and then the Interpreter exits.  All axes are still enabled and active.  
  
There are basically 3 scenarios for continuing:  
  
#2.1 Restart from beginning: in this case the interpreter should have the Restart parameter set so it discards the stopped state and re-initializes from Scratch.  The KMotionCNC Restart button sets things so this will occur.  Loading a new file and such should also do this.  See:  
  
void CKMotionCNCDlg::OnRestart()   
  
#2.2 Continue from where you were.  In this case a check is made to verify the machine is still exactly where it was when we last stopped.  If so it will just continue the job seamlessly.    
  
#2.3 Continue from where you were except the machine was moved to change a broken tool or change some offset.   In this case the libraries detect the machine is not at the same location and prompts the User for a resume sequence to move up to safe z, over, feed down to previous location, etc...  
  
See:  
int CKMotionCNCDlg::LaunchExecution(CString InFile,int begin, int end)