```
float PID_output(float process, float setpoint, float Prop, float Integ, float deriv, int Interval, bool action)
{
float Er;
static float Olderror, Cont, old_Process;
static int Limiter_Switch; // Prevents integral windup
static float Integral;
float derivative;
float proportional;
float deltaT; //Interval time (msec)/1000
float filteredDerivative;
deltaT=float(Interval)/1000;
Limiter_Switch = 1;

if (action==false)
  {
  Er = (process-setpoint); // forward or direct acting
  } else if (action==true)
  {
  Er=(setpoint-process);  //reverse acting is the default for the Temp Control System
  }

//Limiter switch turns integration OFF if controller is already at 100% output or 0% output
//Prevents integral windup, where controller keeps integrating
// when controller output can no longer affect the process.
// 1 is the interval time in seconds

if ((Cont >= 1 && Er > 0) || (Cont <= 0 && Er < 0) || (Integ >= 3600))
       Limiter_Switch = 0;
else
       Limiter_Switch = 1;

Integral = Integral + 100 / Prop / Integ * Er *deltaT * Limiter_Switch; // Integral calculator
derivative = 100 / Prop * deriv * (old_Process-process) / deltaT; // Derivative on process, not error to eliminate derivative action on setpoint
filteredDerivative=DerivativefilterFunction(2, 1,derivative, 1000);
proportional = 100 / Prop * Er;// Proportional calculator

Cont = proportional + Integral + filteredDerivative; // filtering derivative to make it less susceptible to noise
Olderror = Er; // remember previous error for derivative calculator
old_Process=process;
if (Cont > 1) // limit controller output between 0.0 and 1.0 a normalized value
    Cont = 1;

if (Cont < 0)
    Cont = 0;
//*** for display on 20x4 LCD display*****
if(autoTune==false) // no autotune
  {
  lcd.setCursor(9 , 2); //lcd refers to 20x4 LCD display
  lcd.print("    ");
  lcd.setCursor(9 , 2);
  lcd.print((int)(Cont*100.0));
  }
if(component==true & autoTune==false) // no autotune but display PID components
  {
    lcd.setCursor(15 , 0);
    lcd.print("     ");
    lcd.setCursor(15 , 0);
    lcd.print((int)(proportional*100.0));
    lcd.setCursor(15 , 1);
    lcd.print("    ");
    lcd.setCursor(15 , 1);
    lcd.print((int)(Integral*100.0));
    lcd.setCursor(15 , 2);
    lcd.print("     ");
    lcd.setCursor(15 , 2);
    lcd.print((int)(filteredDerivative*100.0));
  }
//****end of display code******
return Cont;
}
```