

Productivity^{3000™} 

Be More Productive

Programming (PID Loop)

Productivity 3000 PAC Programming Example Tutorial

The following information and companion LEARN video have been created to show the ease for which the Productivity 3000 PAC can be programmed. We have chosen to use the PID Loop Instruction as our focus because in most other programmable controllers, it is one of the more difficult functions to understand, program, setup and put into operation. We will spend time in this tutorial getting the viewer up to speed in understanding where and how a PID loop control is typically used, and why it is used.

As a starting point, we cover the operation of the commonly found home heating system that although not commonly controlled by a PID loop controller, makes use of the same basic elements that would be found in a PID application. Next we briefly explain the terms Proportional, Integral and Derivative that are key to how the calculations in our PID loop produce the end results we require. From there we explain our working application example in detail including the components used, how the example is arranged, schematics showing the wiring, using the Productivity 3000's hardware configuration, scaling in engineering units, programming with the built-in PID Loop Instruction, use of a *C-more* Touch Panel to aid in setup and monitoring, and auto-tuning our PID loop. Finally we demonstrate our working example in operation.



Home Forced Air Heating System Controlled by a Thermostat

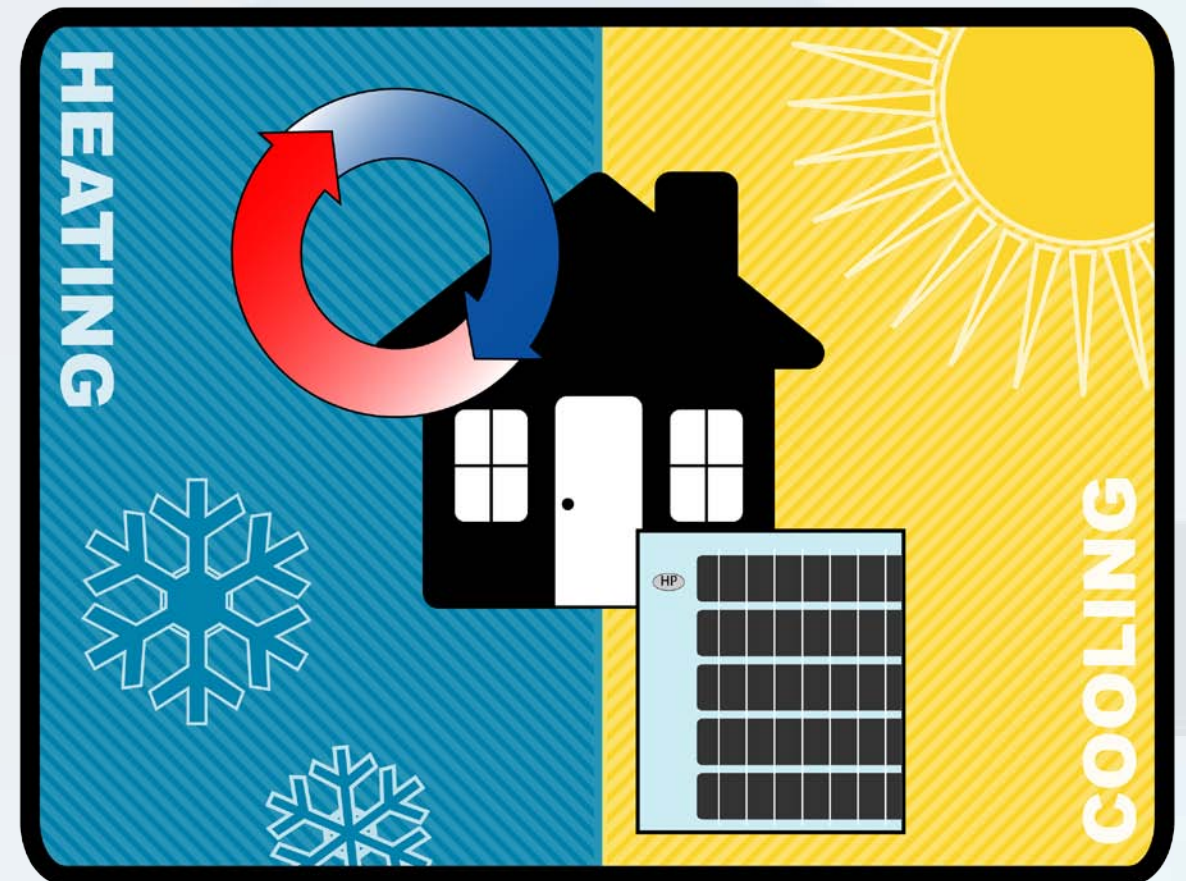
To understand where, how and why a PID loop would be used, think about the typical home forced air heating system that is controlled by a thermostat. This system does not make use of a PID loop controller, but does include many of the same elements, and generally would be considered a basic on/off control type application.

It can be said the home heating system is Closed Loop because it is able to measure the present temperature and base its operation on this value.

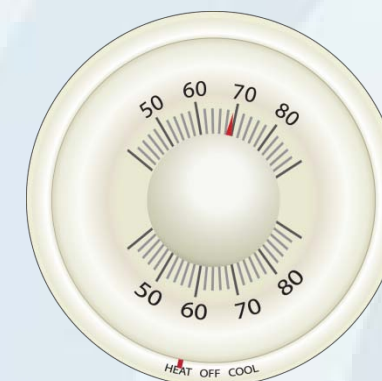
The ability to dial in a desired temperature is our Set Point. Measuring the current temperature becomes our Process Variable. And controlling the heat source becomes our Output.

Older style thermostats include an Anticipator that prevents the room temperature from overshooting. Using an Anticipator is much the same reason we would use a PID loop controller, that is, to keep the end result constant.

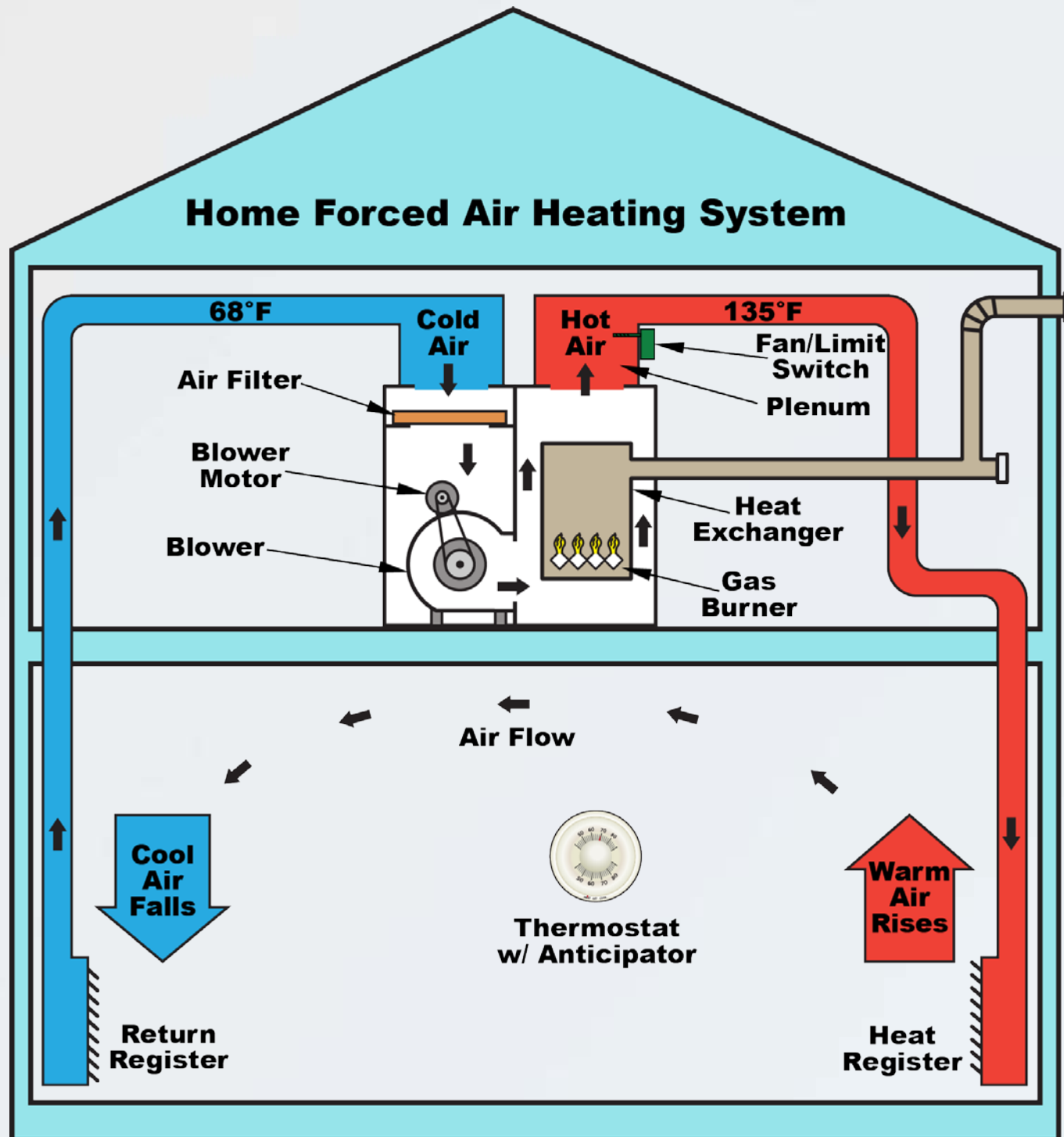
Follow the operational sequence on the next slide!



Thermostat



Home Forced Air Heating System Operational Sequence



- Room temperature falls below thermostat set point and the gas burner is ignited.
- Temperature rises in the plenum and the fan/limit switch turns on the blower when its upper limit is reached.
- Hot air is circulated into the house with the warm air rising.
- Anticipator in the thermostat causes the thermostat to reach the set point before the actual temperature has been reached and turns off the gas burner.
- The blower continues to run until the fan/limit switch falls below the lower limit and the blower is turned off.
- If all conditions are set correctly, actual room temperature should be within a degree or two of set point and the occupant should feel comfortable.

Home Forced Air Heating System Controlled by a PID Loop

Now take a look at the same heating system, but replace the gas burner with an electrical heating element with the ability to vary the amount of wattage (BTUs) that can be produced. As with a gas burner, the heating element is our controlled Output. We still use a Set Point value to let the system know our desired temperature, and we retain the ability to measure the actual temperature, which we call our Process Variable.

Now replace the thermostat with a PID loop control such as the functionality built into the Productivity 3000. What will this do for us? First, using a PID will allow for more accurate control of the room temperature, and second, because we are able to control the amount of heat needed to maintain the Set Point, based on the PID calculations, we likely will use less energy.

In the next series of slides we will briefly cover the functions of the Proportional, Integral & Derivative terms and formula used in our PID loop controller example.

Typical Home Heating/Cooling System



Programmable
Thermostat



Automobile Cruise Control – PID Loop Controlled

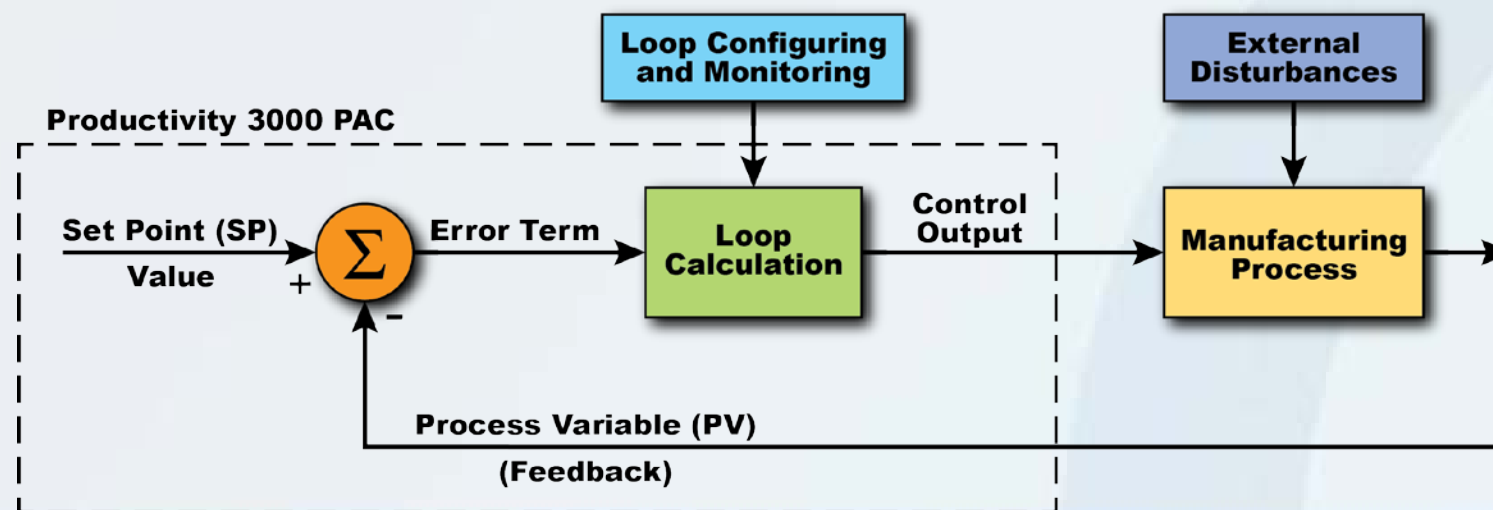
- Obtain your desired speed, press ON to enable the cruise control, and press SET to lock in the current speed of the vehicle. (Set Point)
- Sensor measures current speed and continuously updates. (Process Variable)
- A small computer in the vehicle runs a PID program to calculate and make corrections.
- Any difference between the desired speed and sensed speed becomes an Error which causes the computer to either increase or decrease the vehicle's throttle.
- If a hill is encountered, the throttle is smoothly increased, and a with a downgrade, the throttle is backed off.
- Press ACCEL (+) or DECEL (-) to increase or decrease speed in 1 MPH increments.
- Press OFF or Hit the Brake to disengage the cruise control.



Typical PID Loop Operation

To better understand how a process is controlled with the use of a PID loop controller such as the function that is available in the Productivity 3000, take a look at the key elements shown in the diagram below. The Set Point (SP) value, our desired result, is ‘summed’ with the Process Variable, our feedback. PV lets the control know where the result is in real time. The difference between SP and PV becomes the Error Term. Within the loop calculations, the magnitude of the Error Term value is used with the Proportional gain value to produce a number that will be used to determine how much correction we need from the Output to get the PV closer to the SP. The Proportional gain acts on the present error. The Output correction is further modified by the Integral rate value, which is calculated from the accumulation of past errors. The Derivative rate, when used, is included in the final Output correction based on a prediction of future errors, determined by the current rate of change.

The description and diagram in the next slide represents the PID control scenario for the working example used in this tutorial. In short, the example PID loop control maintains a liquid volume in a tank by measuring the level, converts this value to a volume, and replenishes loss fluid with a variable rate pump.



Proportional, Integral & Derivative

To simplify an explanation on how the PID loop control works, start by looking at just the Proportional calculation. It's result is the error difference between the Set Point value and the Process Variable value multiplied by the value assigned to the Proportional term, also called the Gain. Diagram 1 shows what happens when the Set Point is changed and the Gain value is set too low. The PV will take a long time to get up to the SP. Diagram 2 is an example of having the Gain too high, with the PV overshooting the SP, and the PV going into oscillating. Diagram 3 reflects the ideal result, with the Gain set to have the fastest response time without a large overshoot. Depending on the application and system response, Integral & Derivative values are adjusted to achieve minimum oscillations.

The next slide shows a typical PID formula used for calculating the output based on the three main terms, *Proportional, Integral & Derivative*, that are summed together for the final output correction at any given instance. A more detailed explanation of a PID loop operation can be found at the references listed in the note.

Diagram 1

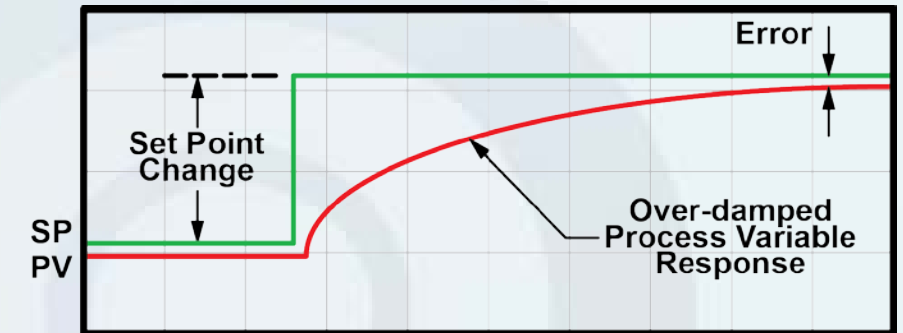


Diagram 2

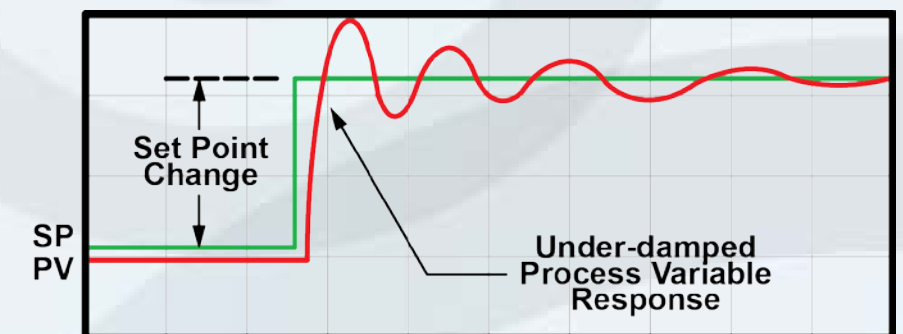
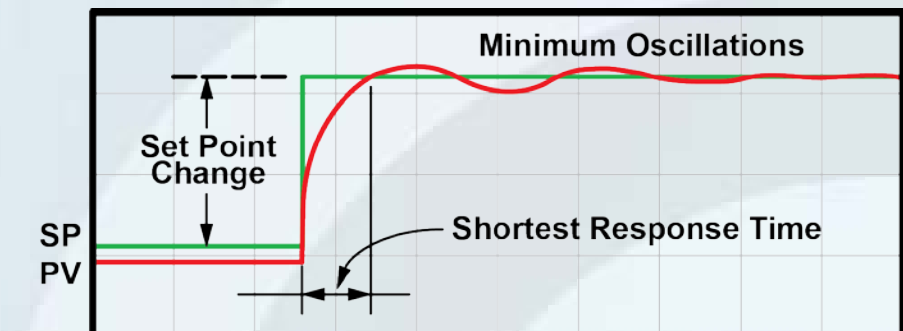


Diagram 3



PID Algorithm Example to Calculate the Control Output *

$$M_n = K_c * e_n + K_i * \sum_{i=1}^n e_i + K_r * (e_n - e_{n-1}) + M_0$$

The diagram shows the PID equation with brackets under each term and labels below them. The labels are: Control Output (under M_n), Proportional Term (under $K_c * e_n$), Integral Term (under $K_i * \sum_{i=1}^n e_i$), Derivative Term (under $K_r * (e_n - e_{n-1})$), and Initial Output (under M_0). A 'Bias Term' label is located below the Integral and Derivative terms, with two arrows pointing upwards to the Integral and Derivative terms respectively.

The PID Algorithm variables and related variables are:

T_s = Sample Rate

K_c = Proportional Gain

$K_i = K_c * (T_s/T_i)$ Coefficient of Integral Term

$K_r = K_c * (T_d/T_s)$ Coefficient of Derivative Term

T_i = Reset Time (Integral Time)

T_d = Rate Time (Derivative Time)

SP_n = Set Point for Sampling Time "n" (SP value)

PV_n = Process Variable for Sampling Time "n" (PV)

$e_n = SP_n - PV_n =$ Error Term for Sampling Time "n"

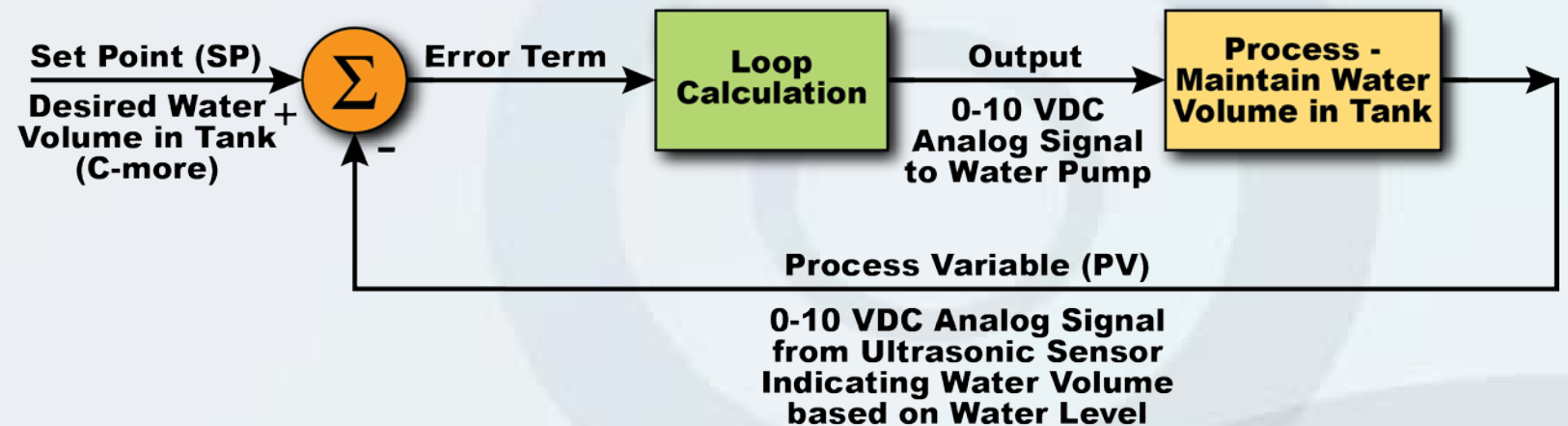
M_0 = Control Output for Sampling Time "0"

M_n = Control Output for Sampling Time "n"

* Note: Additional information about PID Algorithms can be found in the Productivity 3000 online help feature, in many of the *DirectLOGIC* user manuals that are available as a download from the *AutomationDirect* website, and also doing an online web search on 'PID algorithm'.

PID Loop Operation of Working Application Example

- Set Point (SP) – used to set the desired volume in gallons of water required in the water tank.
- PID Loop Calculation – an Error based on the difference between the Set Point and feedback from the Process Variable is used to determine an output correction calculated by the Proportional, Integral and Derivative values.
- Output – this is the calculated result from the loop calculation that determines the water pump rate at any given instance.
- Process Variable (PV) – feedback used to determine how much the actual volume is off from the desired (SP) value.



P, I & D Terms Explanation:

Proportional (P) depends on the *present* error, *Integral* (I) on the accumulation of *past* errors, and *Derivative* (D) is a prediction of *future* errors, based on current rate of change.

PID Loop Application Example Explanation

For the PID loop application used in this tutorial, we will control the volume of a liquid in a water storage tank. Refer to the application diagram shown on the next slide as the process is explained.

A ten (10) gallon cylindrical water tank is used as our main process focus. A sixteen (16) gallon rectangular tank is used as a reservoir. A variable output diaphragm pump is used to fill the process tank from the reservoir. The rate of the pump is 0 to 1 gallon per minute and is controlled by a DC motor controller that will accept a 0-10 VDC analog signal to produce the output pumping rate. The liquid in the process tank drains back to the reservoir tank by means of a siphon tube, and the rate at which it drains is controlled by a manually adjustable ball valve. There is a flow indicator to visibly show the liquid draining.

The ball valve can be adjusted to speed up or slow down the drainage rate.

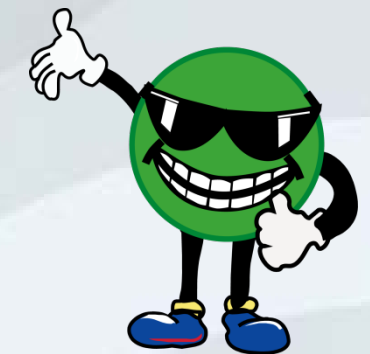
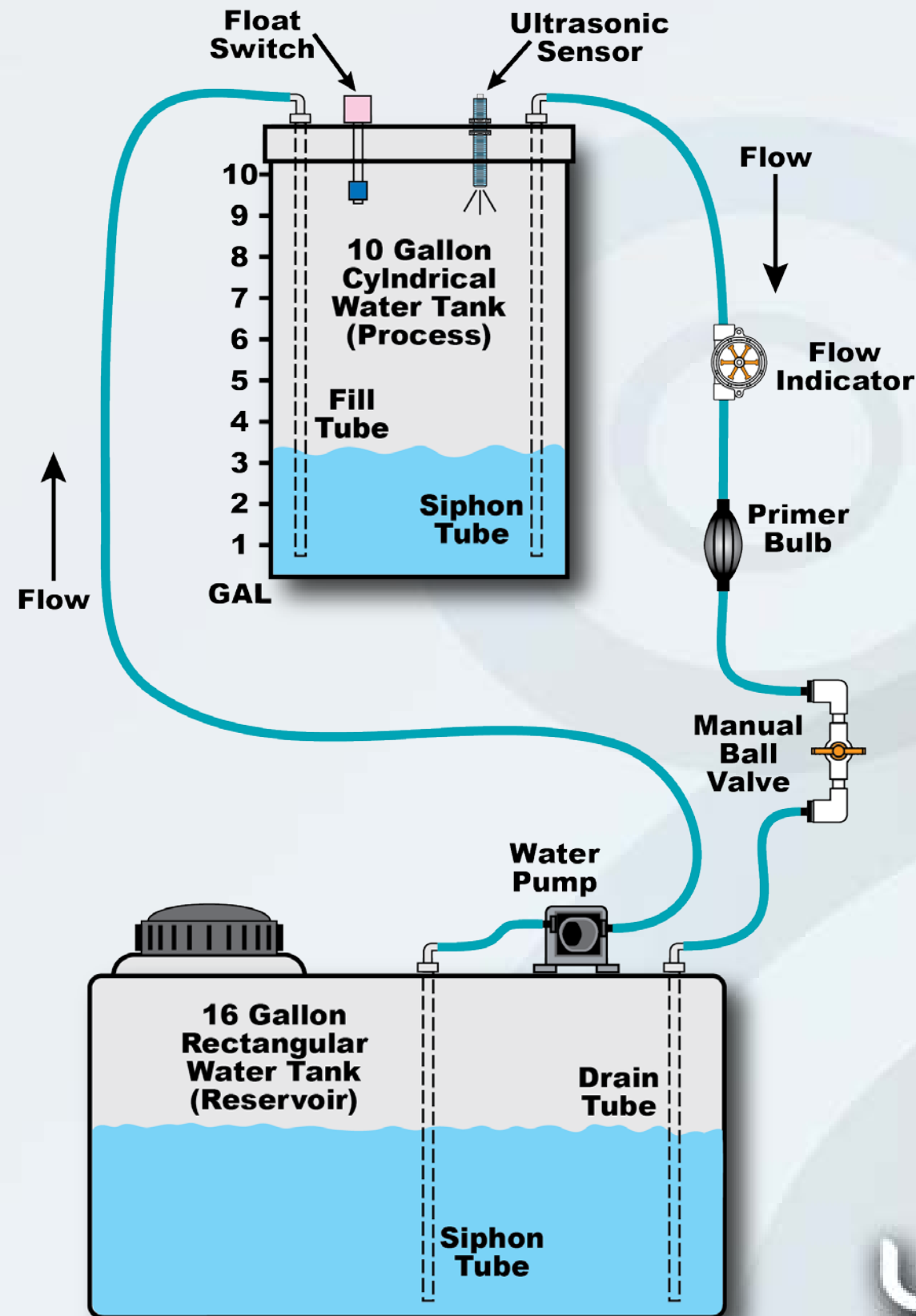
The volume is calculated by determining the level in the storage tank using an ultrasonic probe that produces a 0-10 VDC analog signal in relationship to a 100 to 600 mm distance between the top of the process tank and the current water level.

A hand operated primer bulb, as might be found on an outboard motor boat engine gas tank, is used to initially start the drainage flow, with a siphon tube used to prevent loss of flow once the system is primed.

A fill tube is used in the process tank to reduce ripple effect of the entering liquid that can cause the ultrasonic probe's output to fluctuate.

PID Application Diagram

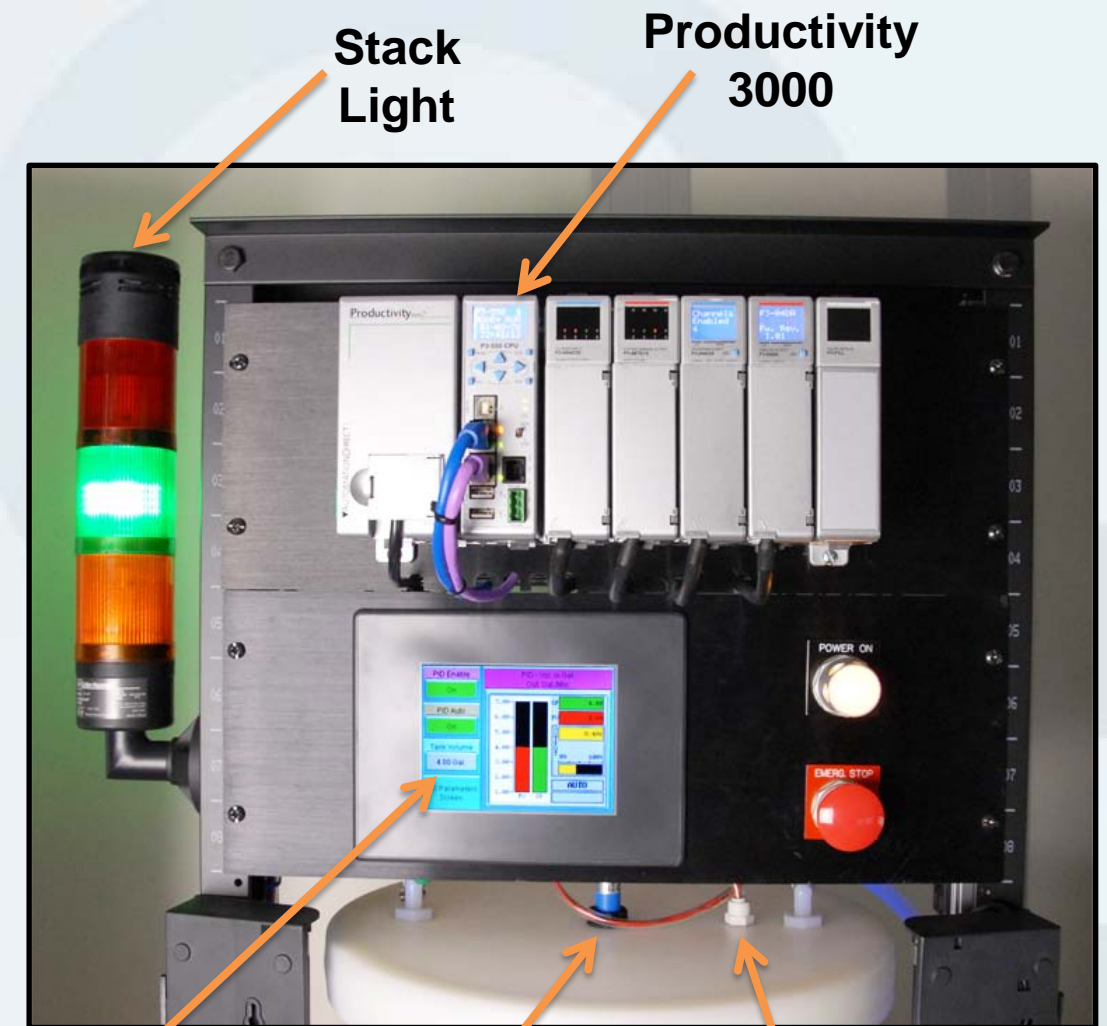
- Process water volume tank.
- Reservoir water tank.
- Pump speed controlled with 0-10 VDC analog output signal from Productivity 3000.
- Ultrasonic distance sensor with 0-10 VDC analog output signal to P3K used to determine water volume based on level.
- Primer bulb to start return flow.
- Manual ball valve used to alter return flow rate.
- Tubes used to eliminate bottom or side penetrations.
- Float switch used as emergency shut down.



PID Loop Application Equipment from ADC

The *AutomationDirect* equipment used for the PID loop example application consists of the following:

- Productivity 3000 PAC (Base, Power Supply, CPU, DC Input, DC Output, Analog Input, and Analog Output Modules)
- *C-more* 6" Touch Panel, TFT, LED Backlight
- *ZipLink* Pre-wired Modules and PLC Cables
- Ultrasonic Sensor, 18 MM, 0-10 VDC Analog Output
- Rhino 24 VDC Power Supplies, DIN Rail Mount
- Relays and Octal Sockets, 75 Series
- Eaton CH Stack Light Tower and Audible Alarm
- Eaton CH 30 MM Pushbuttons
- Single Pole Supplementary Circuit Protectors
- Stride Unmanaged Ethernet Switches
- Ethernet Cat5e Patch Cables



C-more Touch Panel

Ultrasonic Sensor

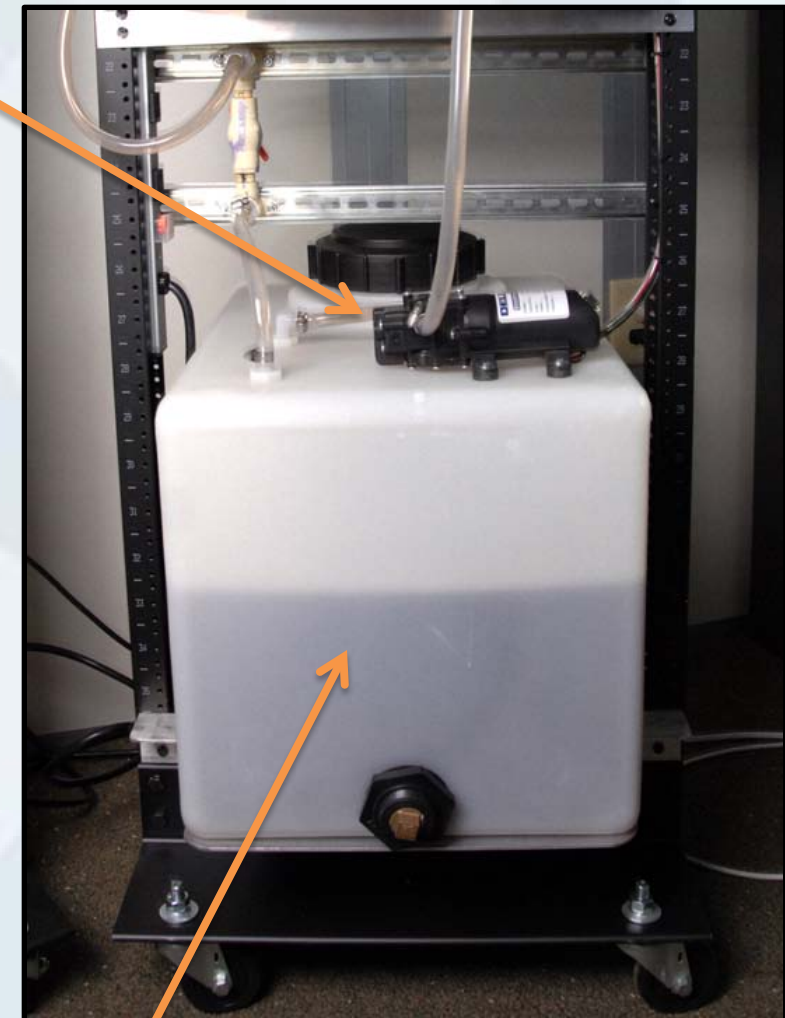
Float Switch

PID Loop Application Equipment from Other Sources

Other equipment used for the PID loop application consists of the following:

- 19" Relay Rack with Casters
- 12 VDC diaphragm pump/motor
- Dart DC motor controller with analog speed input
- GEMS Float switch for overflow safety shutdown
- Primer bulb to start drainage flow from siphon tube
- Flow indicator with paddle wheel to show drainage flow
- Polyethylene tanks, 10 gallon cylindrical for process and 16 gallon rectangular used as reservoir
- Manual ball valve to control drainage flow rate
- Vinyl tubing and fittings

Diaphragm Pump/Motor



16 gal. Reservoir Tank

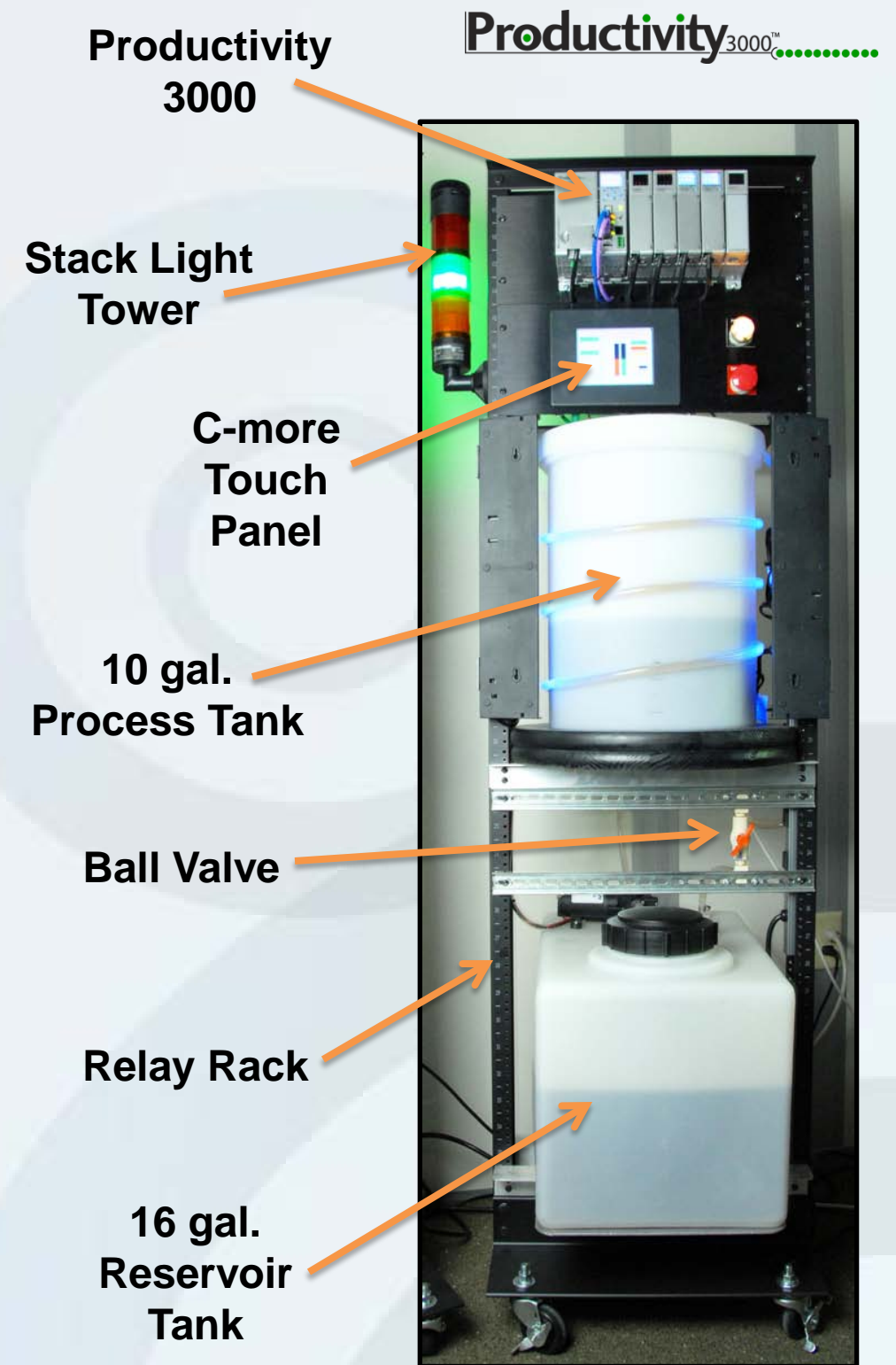
Equipment Arrangement

The entire PID loop example application was designed to fit on a standard 19" relay rack with casters to allow portability.

On the front side, as seen in the layout on the next slide, are the Productivity 3000, *C-more* Touch Panel, Power/E-stop push buttons, process tank with ultrasonic level sensor & overflow flow switch, and Stack Light tower.

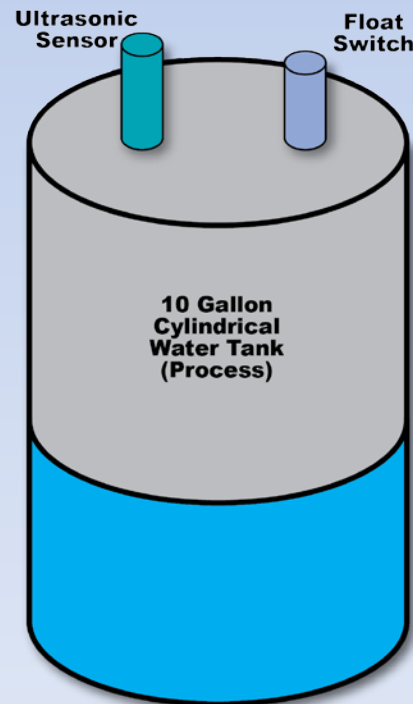
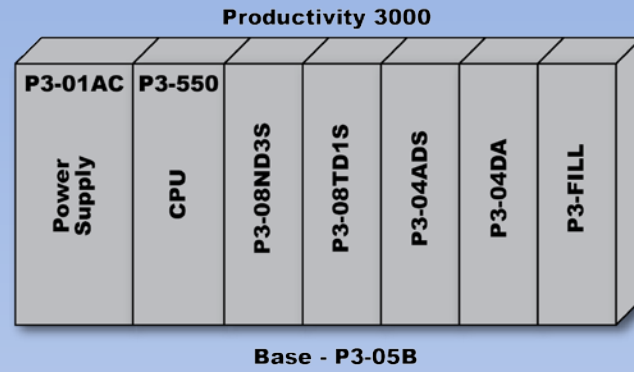
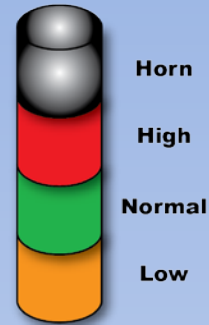
On the rear are found the ZipLink modules, power supplies, Stride Ethernet switches, relays, circuit protection, DC motor controller, reservoir tank and diaphragm pump with 12 VDC motor.

The rear side also includes terminal blocks and wire duct to facilitate wiring. The use of the ZipLink modules and cables make for a quick job and wiring neatness. Notice the ZipLink modules are placed in the same left to right order as the actual Productivity 3000 I/O modules on the front of the assembly. An I/O module layout order of DC input, DC output, analog input & analog output was chosen, and with the ability to cross the ZipLink cables over, the same module order was kept on the rear of the assembly eliminating wiring and troubleshooting confusion.



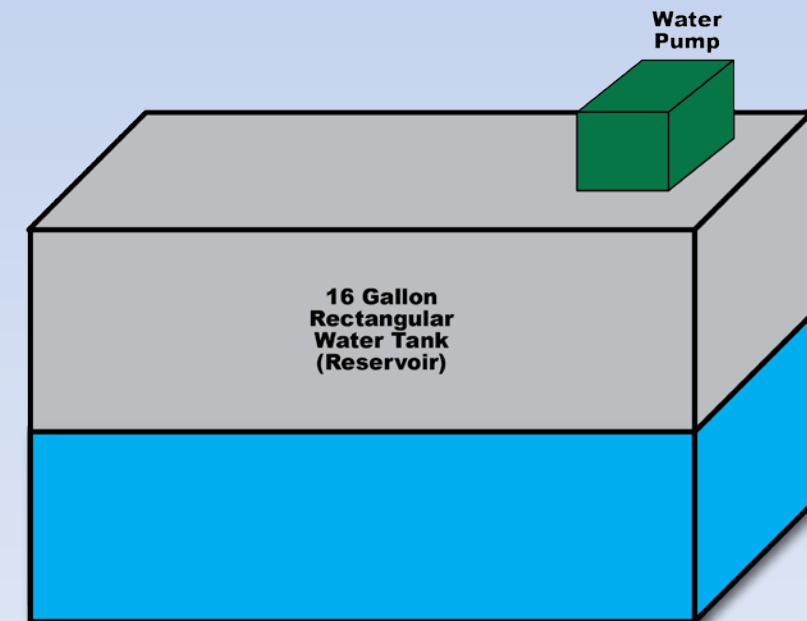
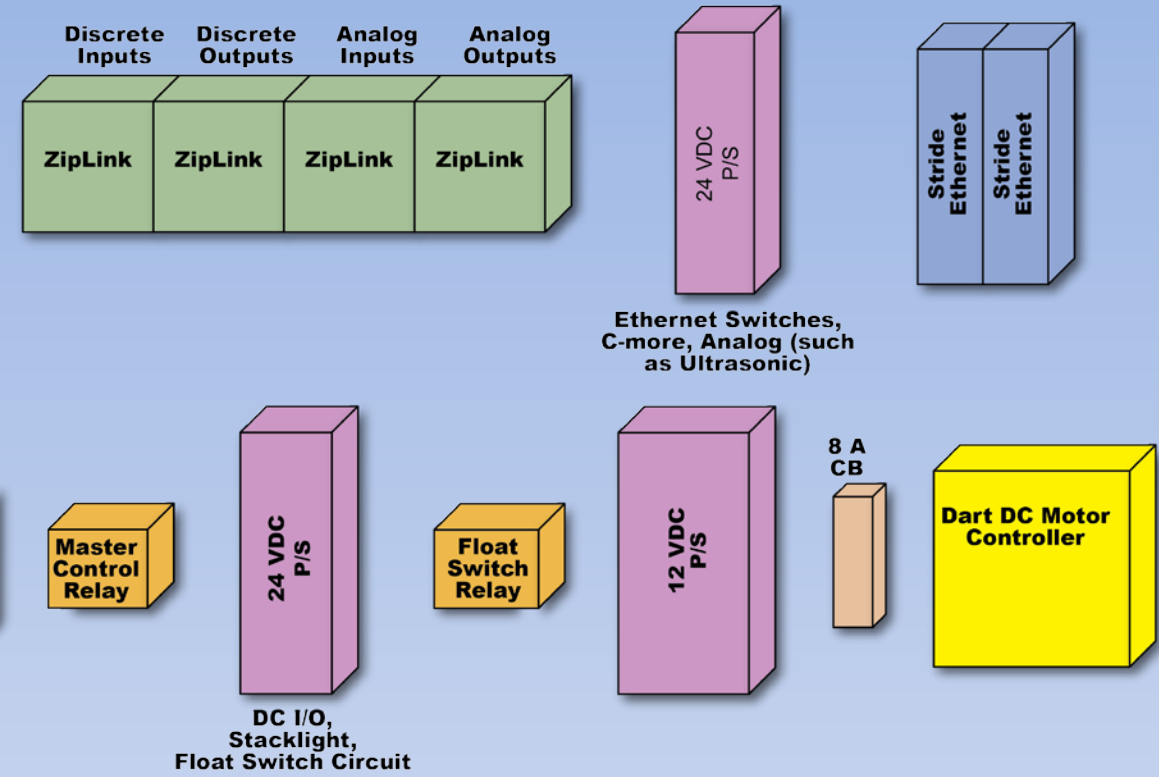
P3K PID Demo Layout

Stacklight



FRONT VIEW

Receptacle Strip



REAR VIEW

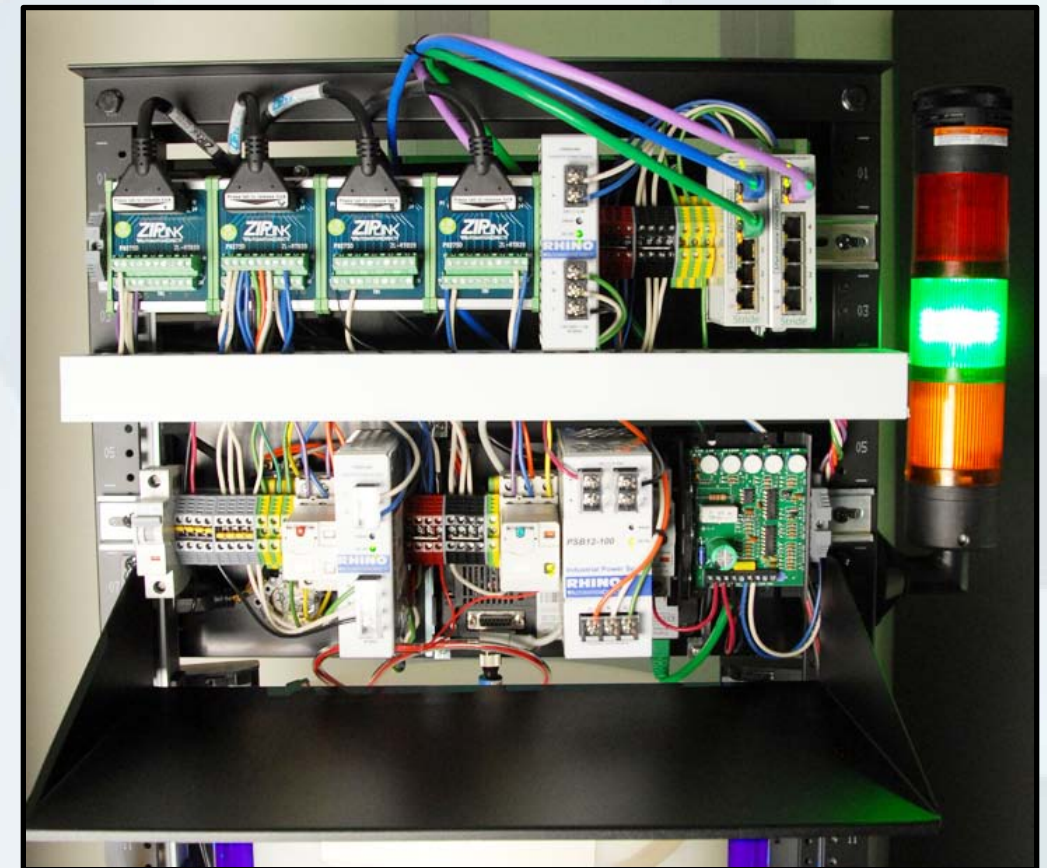
PID Loop Application Equipment Wiring (Schematics)

The next series of four slides cover the schematics used for the PID loop application wiring. The Productivity 3000 Programmable Automation Controller is the heart of the control system by providing the PID functionality required.

Care was taken to use protective devices for the various circuits, and it should be noted that two different 24 VDC power supplies were used to help isolate the analog circuitry from the general DC device circuitry. Also the use of a separate 12 VDC power supply provides power to the Dart DC motor controller that operates the diaphragm pump motor.

A Power On/E-Stop push button and relay circuit was included as a safety to stop all action in case of an emergency, or if the float switch is actuated due to an overflow condition.

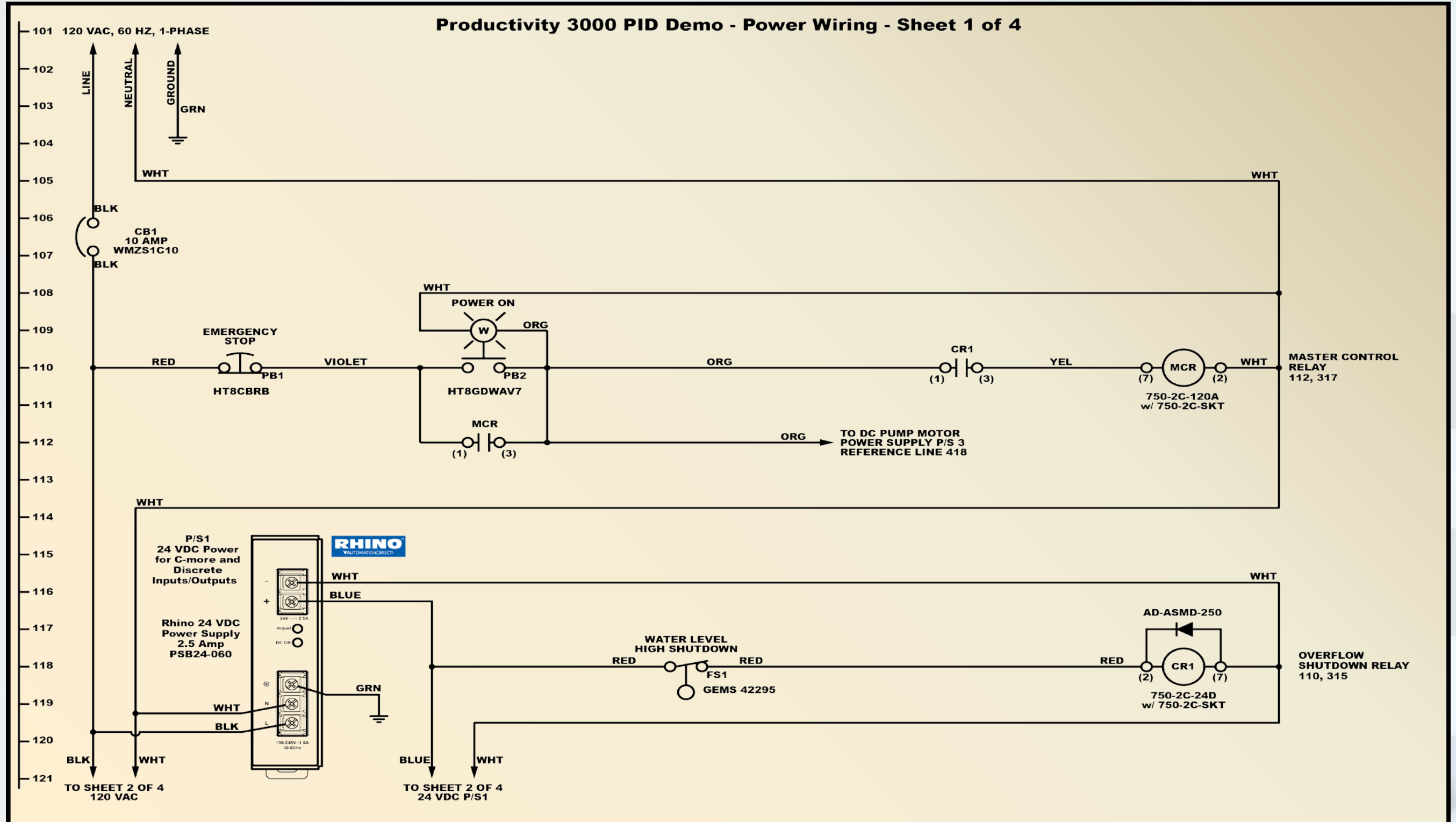
The Productivity 3000 and the *C-more* Touch Panel are networked together using Stride Ethernet switches.



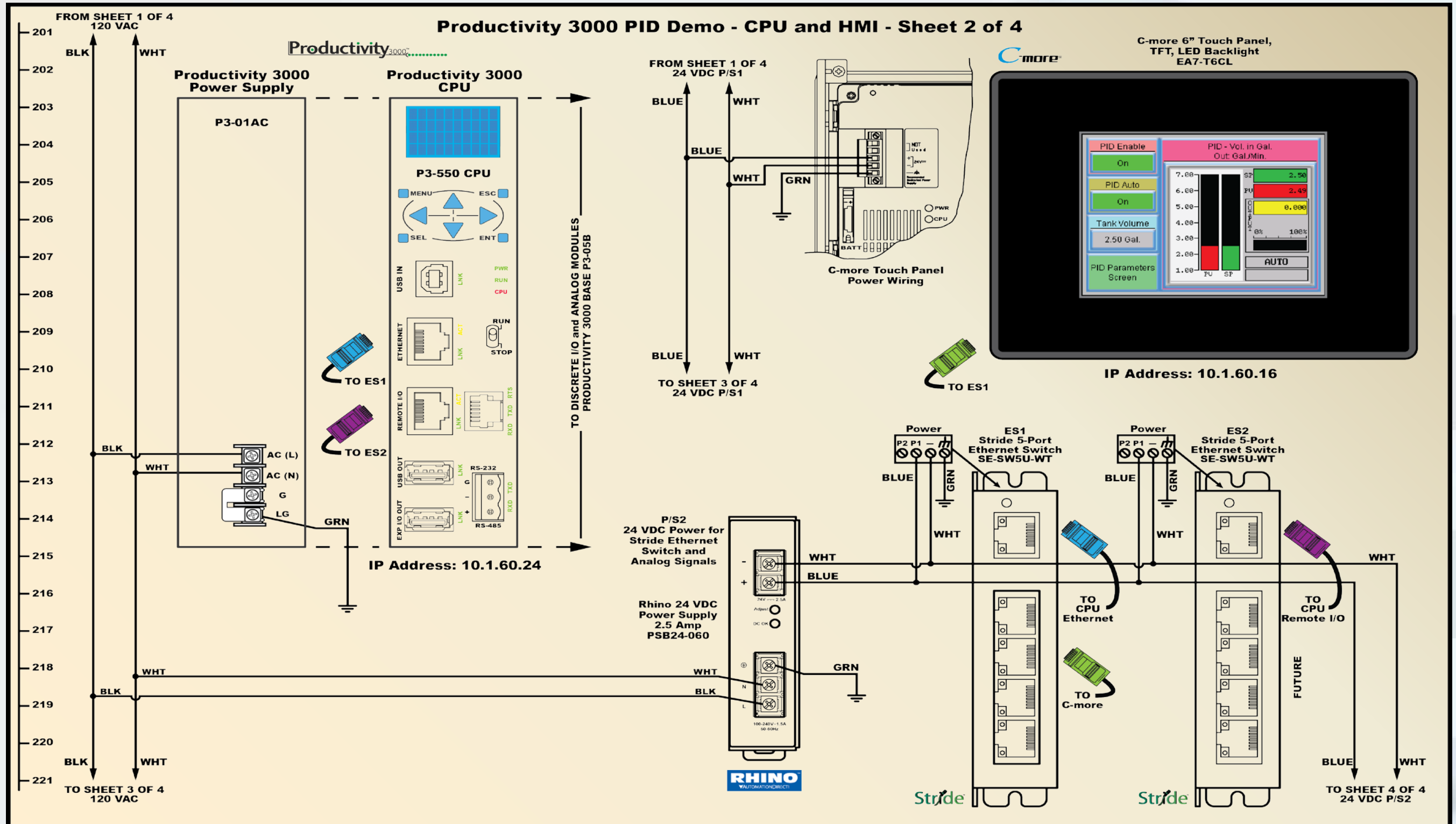
Rear Side Wiring

Schematic Diagram – Power Wiring – Sheet 1 of 4

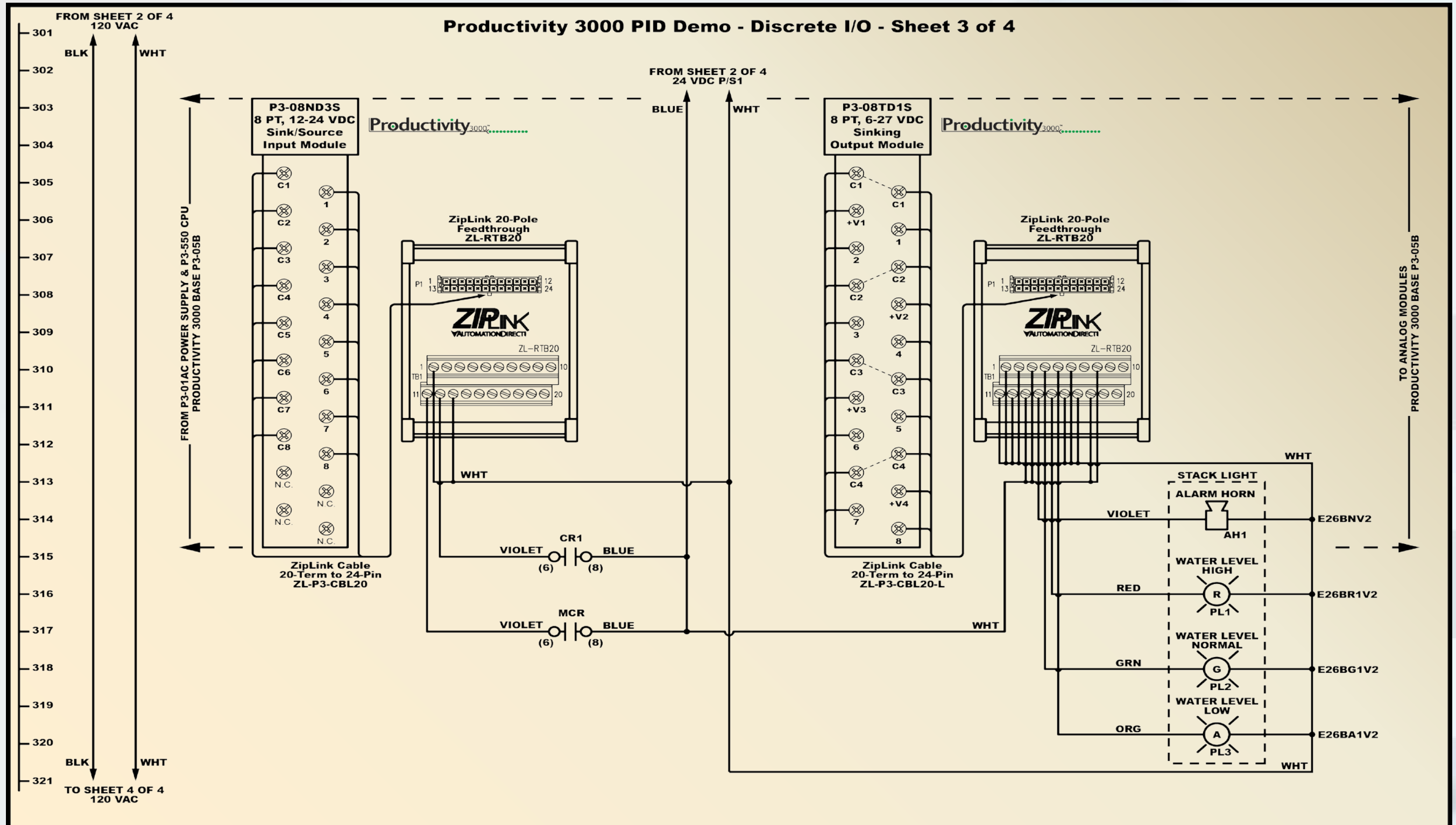
Productivity 3000 PID Demo - Power Wiring - Sheet 1 of 4



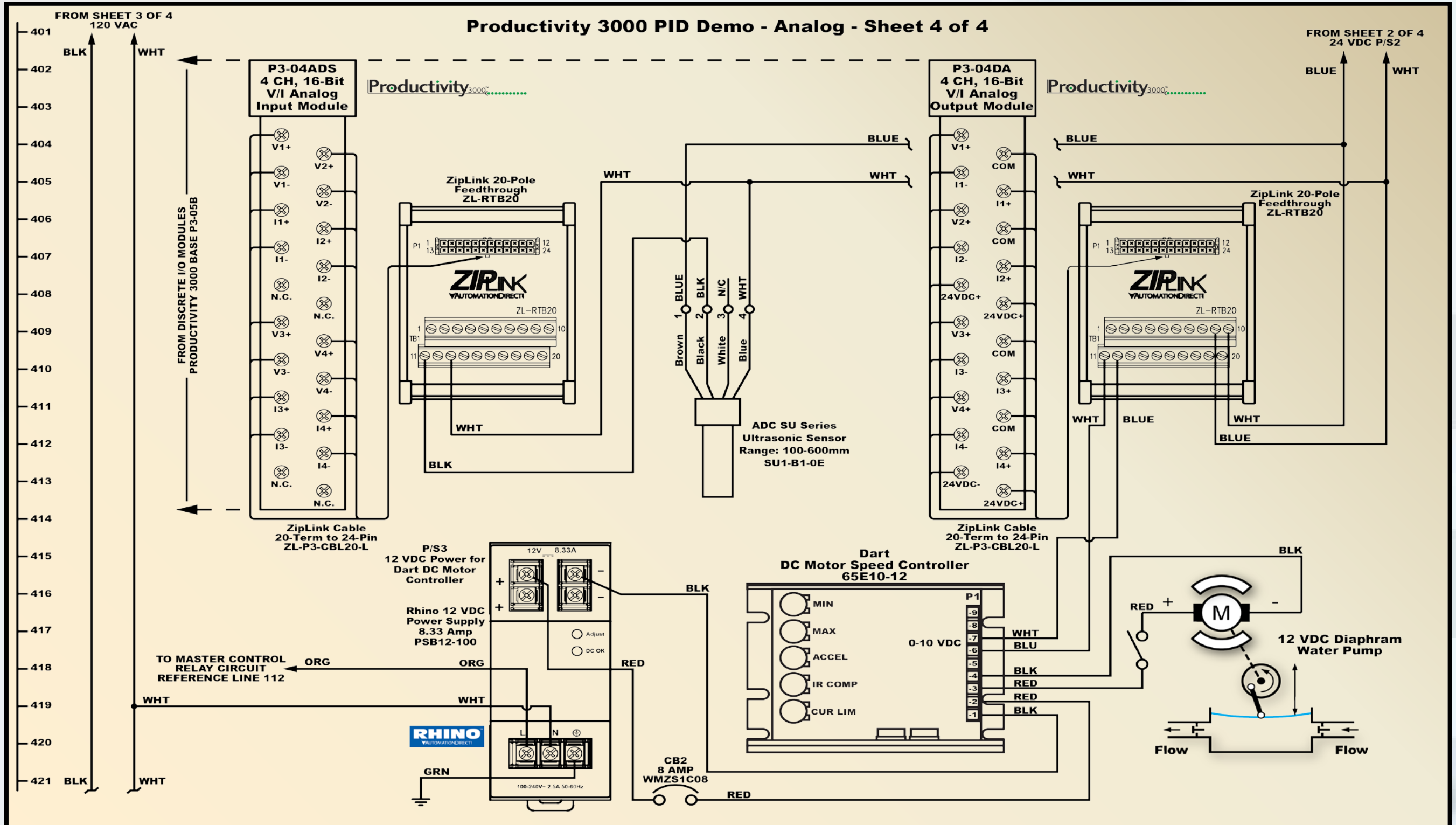
Schematic Diagram – CPU & HMI – Sheet 2 of 4



Schematic Diagram – Discrete I/O – Sheet 3 of 4



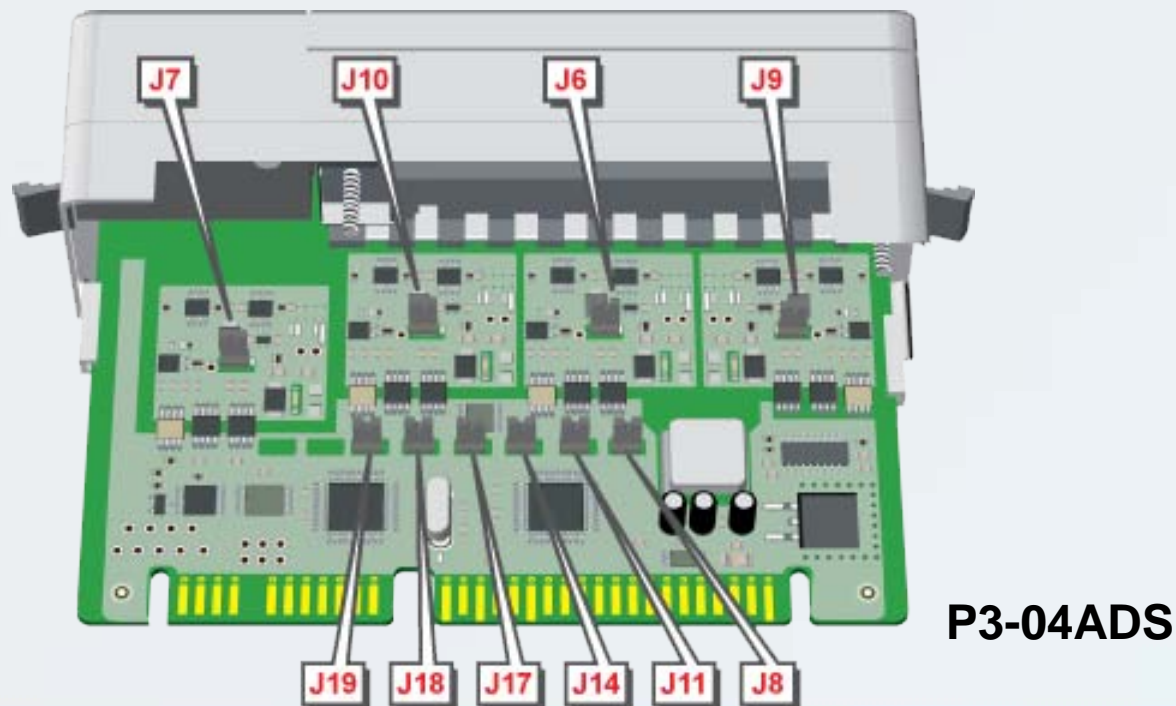
Schematic Diagram – Analog Inputs & Outputs – Sheet 4 of 4



Analog Input Module Jumper Settings

The analog input module used for the PID Loop application has the ability to be configured for different input current and voltage ranges, such as 0-5 VDC, 0-10 VDC, 4-20 mA, etc.

The Ultrasonic Sensor we are using has an analog output of 0 to 10 VDC. The P3-04ADS Analog Input Module is setup so that all channels will accept a 0-10 VDC signal. This will provide a 0-65,535 count range.



P3-04ADS Jumper Orientation										
J19	J18	J17	J14	J11	J8	J7	J10	J6	J9	Function
N	N	-	-	-	-	-	-	-	-	Enable channel 1
Y	N	-	-	-	-	-	-	-	-	Enable channel 1 & 2
N	Y	-	-	-	-	-	-	-	-	Enable channel 1, 2 & 3
Y	Y	-	-	-	-	-	-	-	-	Enable all channels
-	-	N	N	-	-	Y	-	Y	-	Range 0-5V for channels 1 & 3
-	-	Y	N	-	-	N	-	N	-	Range 0-10V for channels 1 & 3
-	-	N	Y	-	-	N	-	N	-	Range +/-10V for channels 1 & 3
-	-	Y	Y	-	-	Y	-	Y	-	Range 0-20mA for channels 1 & 3
-	-	-	-	N	N	-	Y	-	Y	Range 0-5V for channels 2 & 4
-	-	-	-	Y	N	-	N	-	N	Range 0-10V for channels 2 & 4
-	-	-	-	N	Y	-	N	-	N	Range +/-10V for channels 2 & 4
-	-	-	-	Y	Y	-	Y	-	Y	Range 0-20mA for channels 2 & 4

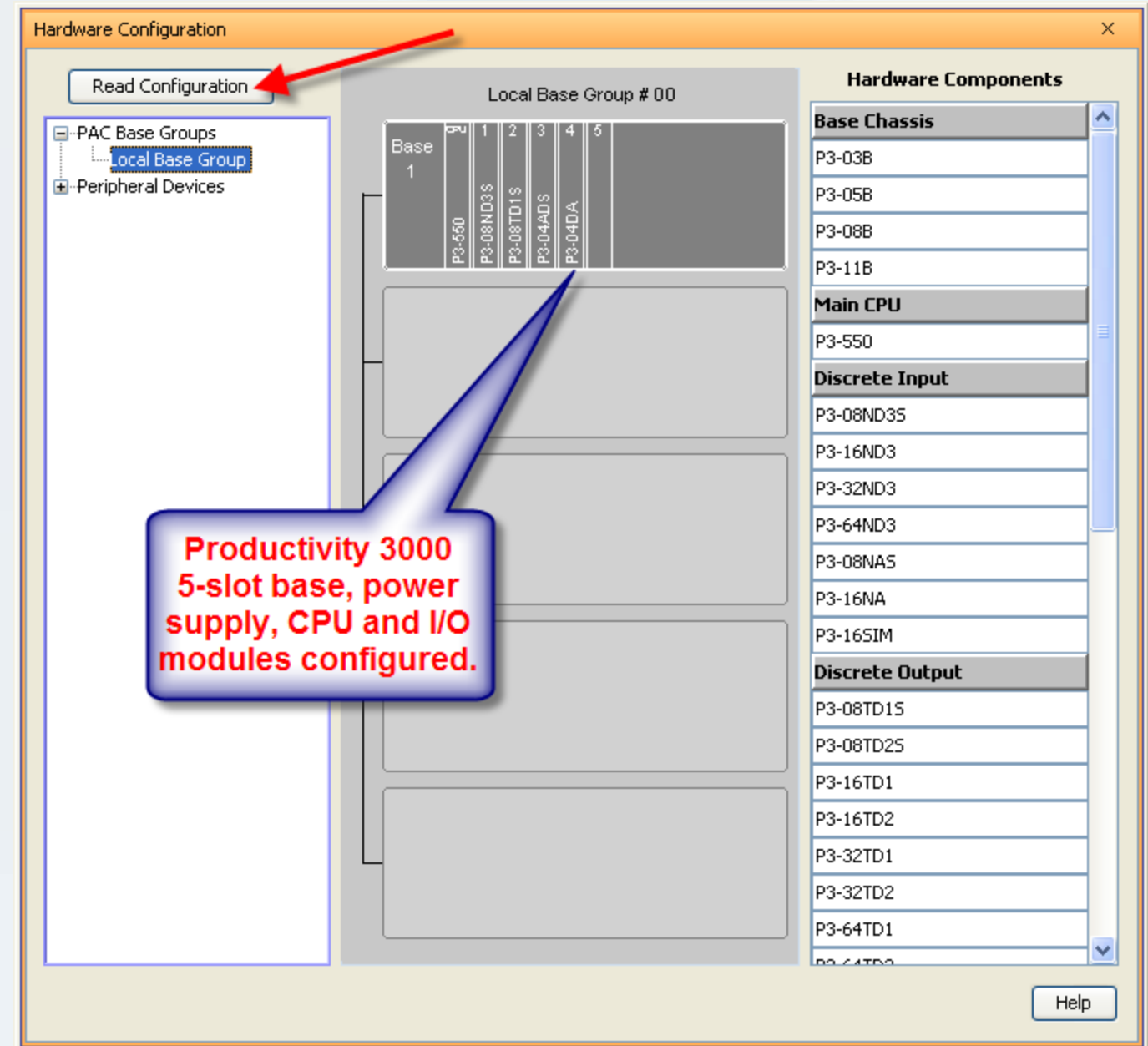
Legend: N = No jumper installed (open)
Y = Jumper installed

Hardware Configuration

The first step we need to accomplish within the Productivity 3000 programming software is to configure the PAC system hardware. With the programming software installed on a personal computer, connect to the P3K system via USB or Ethernet. Open the Hardware Configuration dialog under the Setup Application Tools tab. Click on the Read Configuration button as shown to the right. (The CPU needs to be in Stop mode when reading the hardware configuration.) After a few moments, all of the attached I/O modules, and any other attached hardware, will be recognized.

View the LEARN video “Part 2 – How to Configure the Productivity 3000 Programmable Controller” for addition details on Hardware Configuration.

<http://learn.automationdirect.com/PLCs.html>



Hardware Configuration

Tagnames and Tagname Database

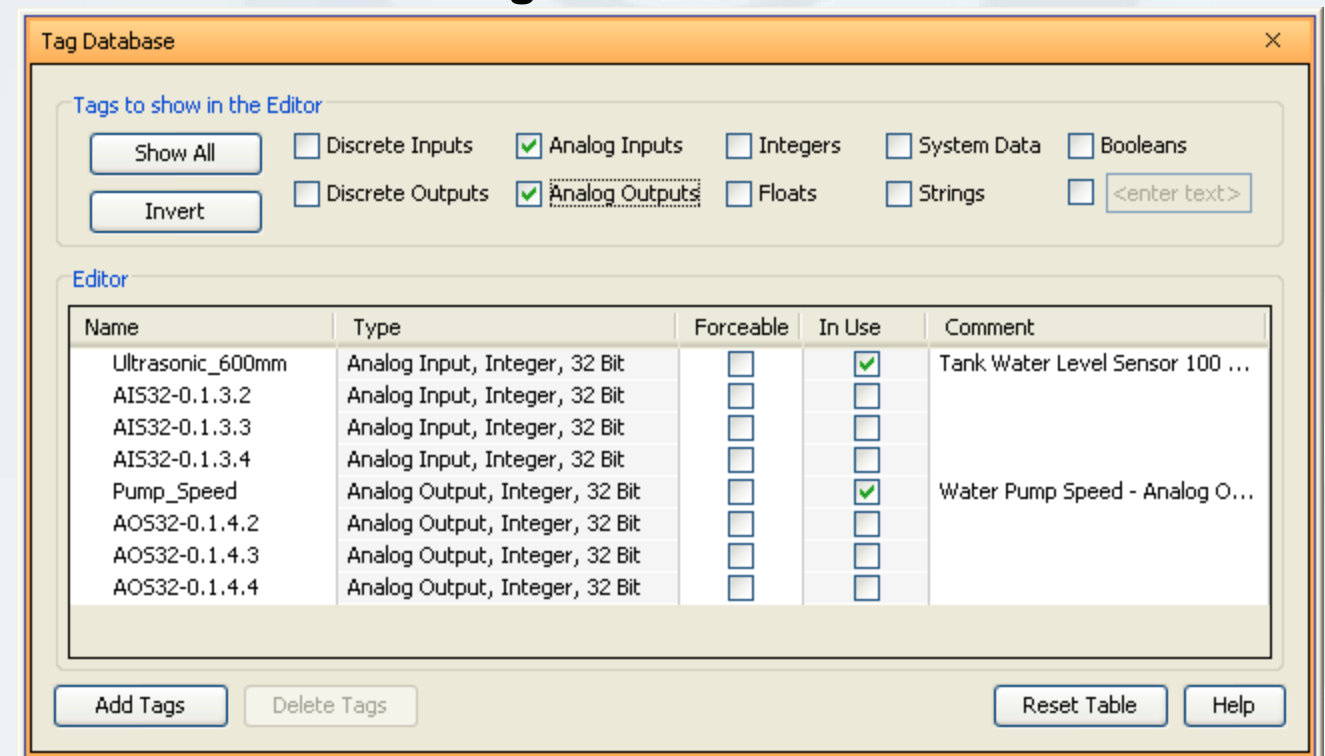
Before we get into the actual programming, it will be helpful to have an understanding of Tagnames and the Tagname Database. Their use will definitely make programming, program structure, and later troubleshooting easier.

The Productivity 3000 is a “Tagname Based” Programmable Automation Controller. It allows the end user to create descriptive names (Tagnames) for all of the variables and I/O points which makes programming go quicker. Tagnames have the freedom to be defined as any data type instead of using arbitrary blocks of wasted memory for data types not required.

Physical I/O points can have descriptive names (Tagnames) instead of numerical base/rack/module/point references that are the normal default.

And to save time, the Tagnames can be exported from the Productivity 3000 software and then imported into the *C-more* programming software to both save time and prevent typos.

Tagname Database



PID Analog Input Calculating & Scaling Tank Volume – 1 of 4

Using a PID loop is always easier if you are able to use the Process Variable values in familiar engineering units. The Productivity 3000 programming software includes various scaling and math instructions that make this task simple.

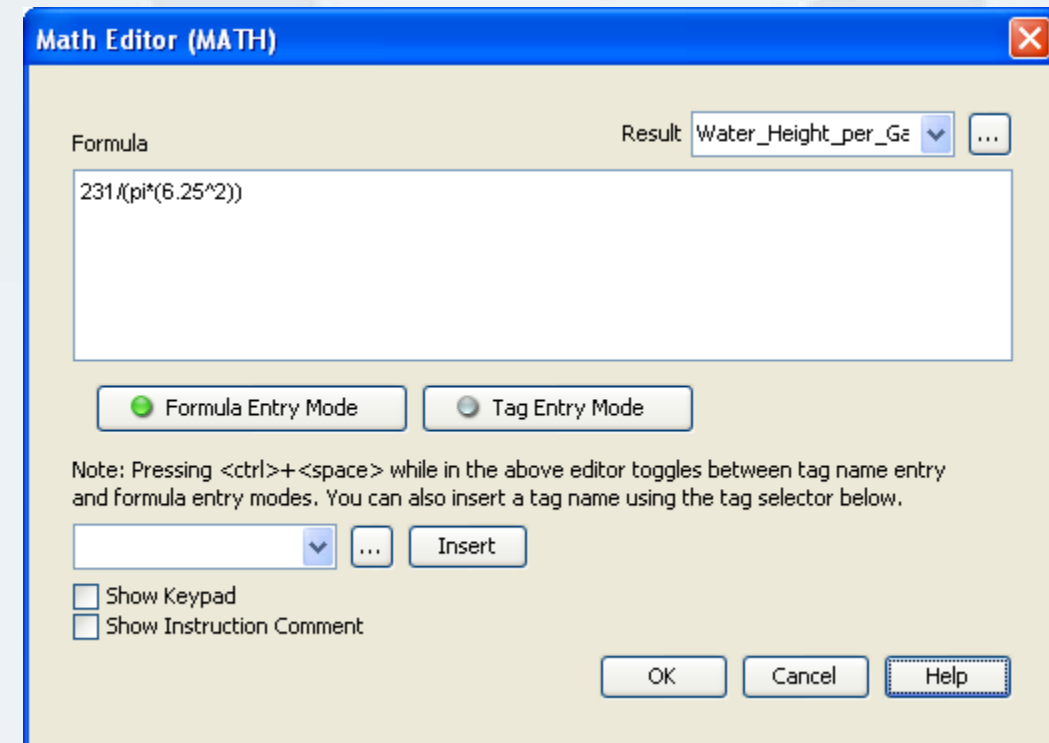
In our application we need to control the water volume in a 10 gallon tank. We are using an ultrasonic sensor that can read a distance of 100 to 600 mm from the front face of the sensor with an analog output of 0 to 10 VDC. Converting metric to inches, the water level when closest to the sensor would be 3.94 inches and furthest would be 23.62 inches. Keep in mind the volume relationship to sensor distance is inverted, with the most volume being at the closest distance.

Reference: One (1) gallon = 231 cubic inches.

The inside diameter of the 10 gallon tank measures 12.5 inches.

“Water_Height_per_Gallon” equals:
 $231 / (\text{Pi} * (\text{radius squared})) =$
 $231 / (3.14159 * (6.25 \text{ sq.})) = 1.88 \text{ inches}$

(cont'd)



Math Instruction

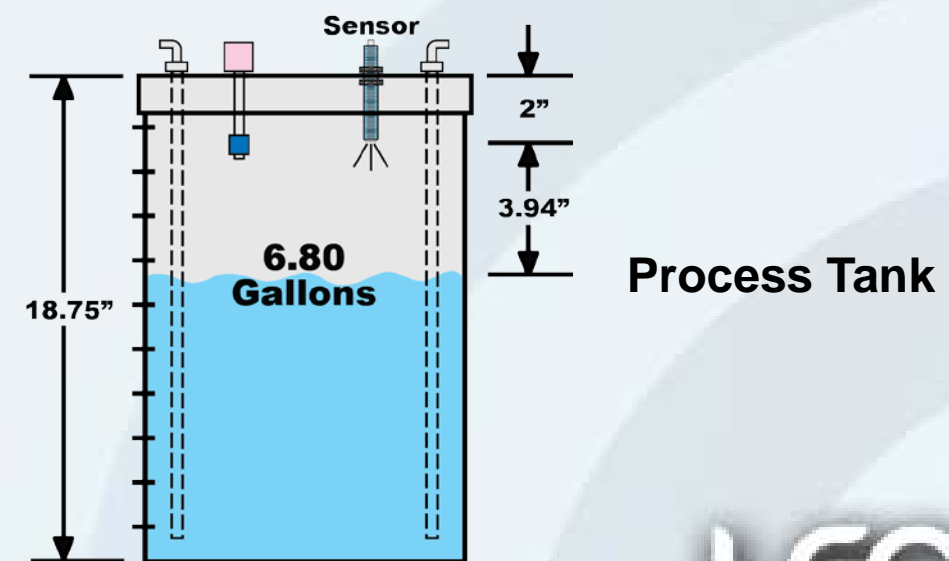
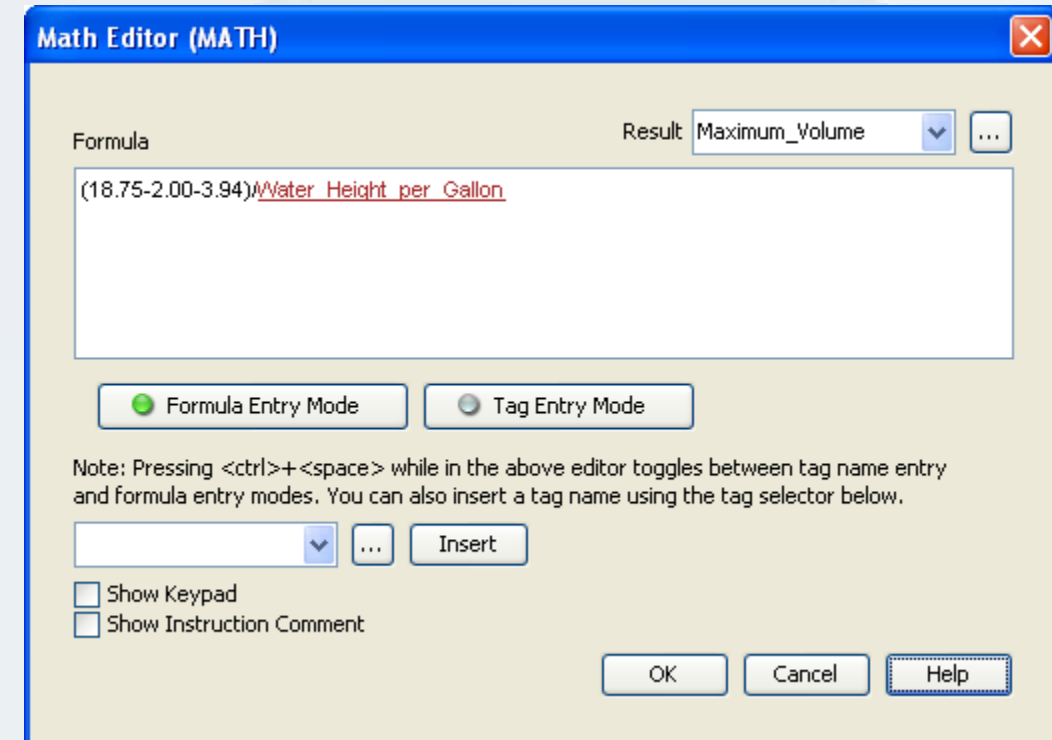
PID Analog Input Calculating & Scaling Tank Volume – 2 of 4

“Maximum_Volume” equals:

Total tank height of 18.75 inches, minus sensor face at 2 inches into tank, minus minimum sensor distance at 3.94 inches = 12.8 inches divided by 1.88 inches height per gallon = 6.80 gallons.

The closest distance the Ultrasonic Sensor can measure is 3.94 inches (100 MM) from the face of the sensor, and the sensor is mounted through the process tank’s lid a distance of 2 inches. This is a total of 5.94 inches, subtracted from the tank’s total height of 18.75 inches would be at 12.81 inches from the bottom of the tank. Because each 1.88 inches of height in the tank represents 1 gallon of water, the highest level that can be read by the sensor will be 12.81 divided by 1.88 inches height per gallon = 6.80 gallons. (cont’d)

Math Instruction



PID Analog Input Calculating & Scaling Tank Volume – 3 of 4

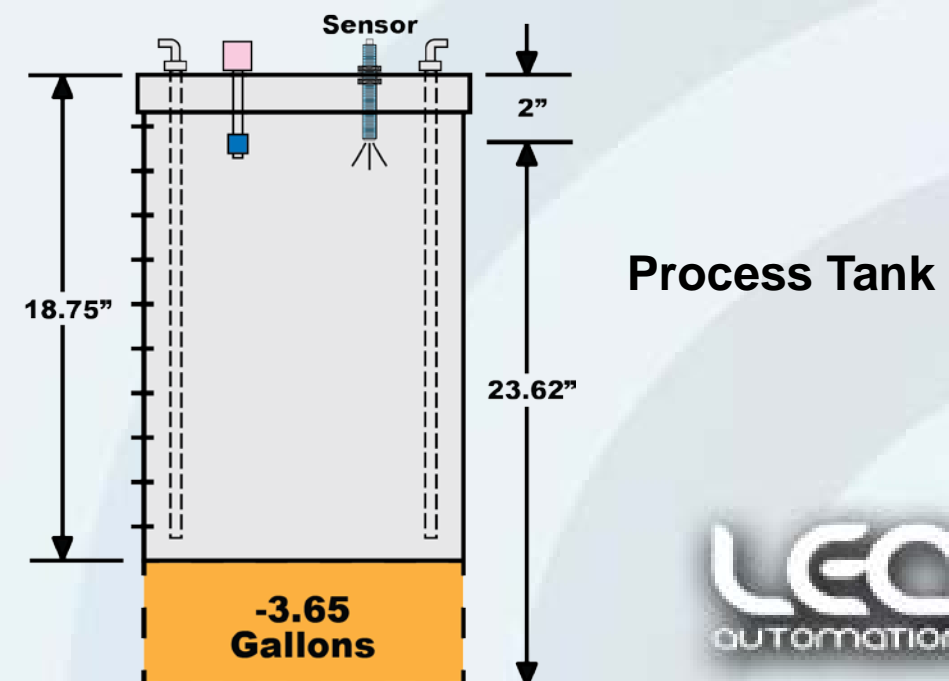
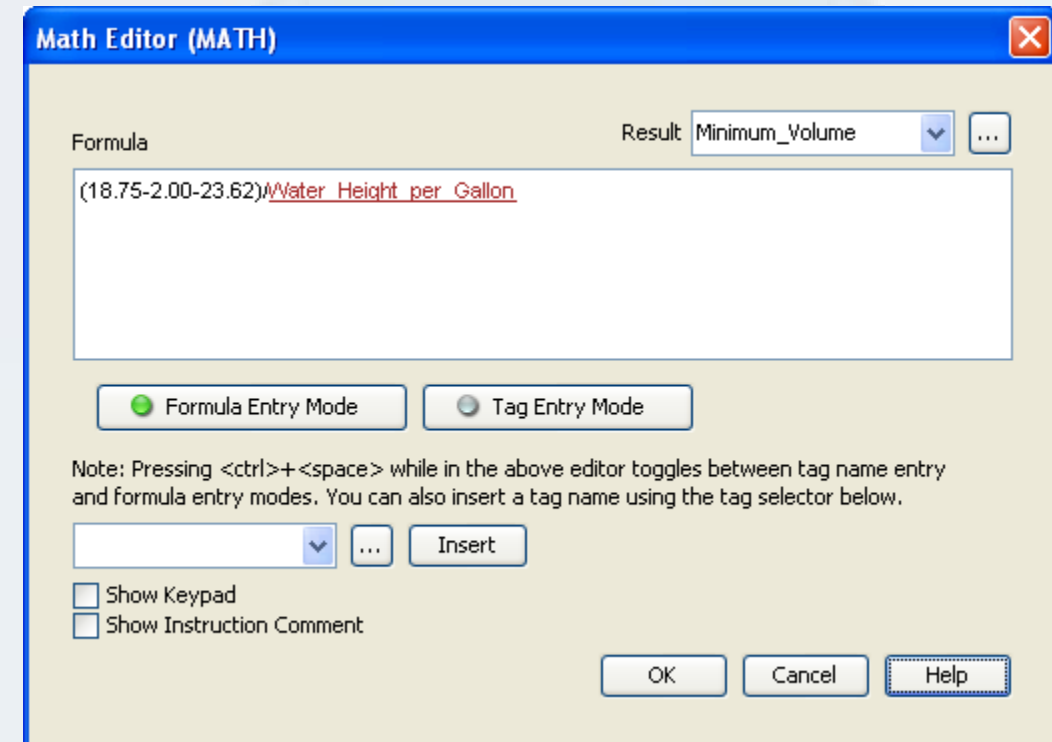
“Minimum_Volume” equals:

This calculation takes a little more thought in what is going on here. The Ultrasonic sensor can measure out to a distance of 23.62 inches. Since the process tank has a height of 18.75 inches, the sensor could actually see past the bottom of the process tank if there were no obstructions, but we want to keep the final volume valve at zero when the tank is empty, and still use the total 0 to 65,535 count range of the analog input module to maintain our resolution.

Total tank height of 18.75 inches, minus sensor face at 2 inches into tank, minus maximum sensor distance at 23.64 inches = “negative” 6.94 inches divided by 1.88 inches height per gallon = “negative” 3.65 gallons.

(cont'd)

Math Instruction



PID Analog Input Calculating & Scaling Tank Volume – 4 of 4

The final task for the Analog Input from the Ultrasonic Sensor is to use the Scale (Linear) Instruction to provide our tank volume in engineering units of gallons of water.

Use the Hardware Configuration to assign the Tagname “Ultrasonic_600” to the first channel input (AIS32-0.1.3.1) on the P3-04ADS analog input module.

“Tank_Volume” equals:

Input Min from “Ultrasonic_600mm” = 0

TO

Input Max from “Ultrasonic_600mm” = 65,535

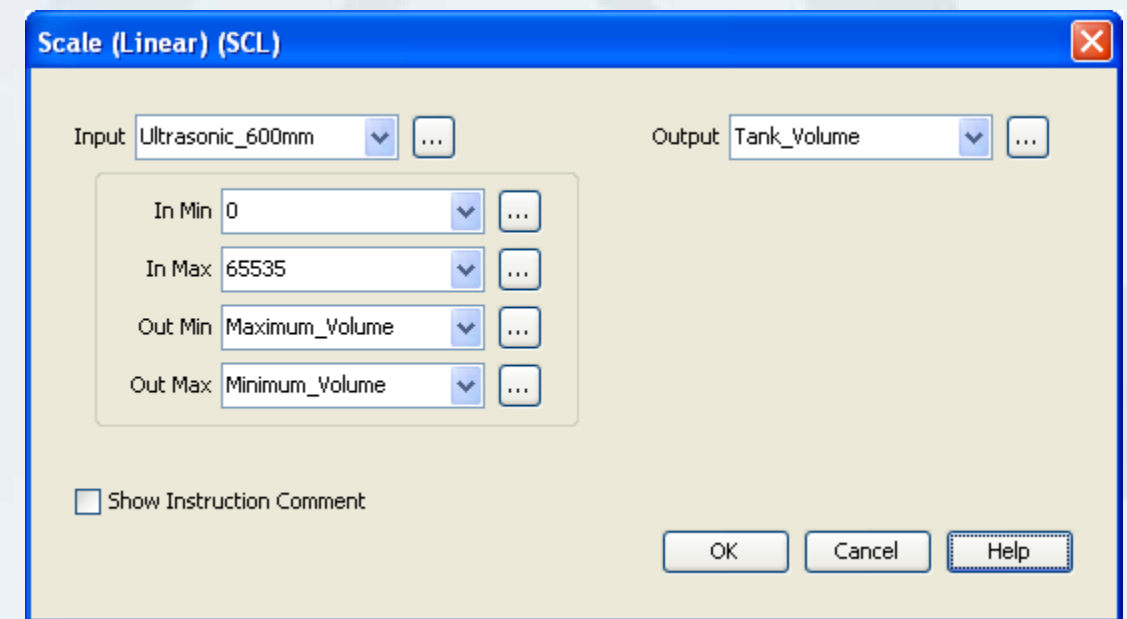
PRODUCING:

Output Min from “Maximum_Volume” (6.80 gal.)

TO

Output Max from “Minimum_Volume” (-3.65 gal.)

Again, maximum volume is seen when the water level is closest to the face of the Ultrasonic Sensor.



Scale (Linear) Instruction

PID Analog Output

The 12 VDC diaphragm motor/pump is capable of producing a rate up to 1 gallon per minute of liquid transfer.

With the use of the Dart DC motor controller, we can vary the pump rate from 0 to 1 GPM by providing a 0 to 10 VDC analog signal to the DC motor controller.

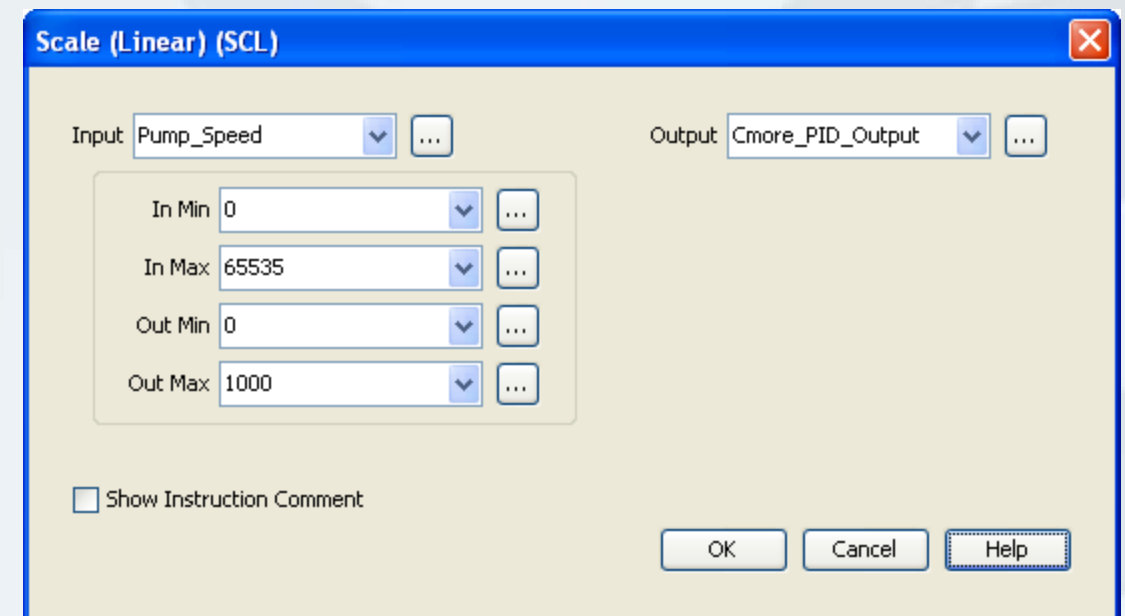
0 to 10 VDC is represented by a count value of 0 to 65,535.

As will be seen in the PID Loop Instruction that follows, we can use the PID's Process Output directly to control the pump rate to produce the maximum resolution without the need to scale. The tagname for the process output is labeled "Pump_Speed".

But, it would be nice to have the *C-more* Touch Panel display the actual pump rate in GPM.

We use a Scale (Linear) Instruction to take the "Pump_Speed", scale it to a 0 to 1,000 value, and give it a tagname called "Cmore_PID_Output".

Within the *C-more*, 3 decimal places to the left are used to display 0 to 1.000 GPM.



Scale (Linear) Instruction

PID Loop Instruction – 1 of 3

Create the PID loop by dragging and dropping the PID Loop instruction in the output portion of the ladder rung. Fill in the Loop Name and Tag Names.

- Loop Name = “Water Volume Control”
- Set Point = “Tank_Volume_SP”
- Process Variable = “Tank_Volume”
- Input Range Max = “Maximum_Volume”
- Input Range Min = “Minimum_Volume”
- Process Output = “Pump_Speed”
- Output Range Max = 65,535
- Output Range Min = 0
- Output Upper Limit = 65,535
- Output Lower Limit = 0
- Proportional (Gain): “Proportional”
- Integral (Reset) Time: “Integral” (Sec.)
- Derivative (Rate) Time: “Derivative” (Sec.)

PID Loop Instruction

(cont'd)

PID Loop Instruction – 2 of 3

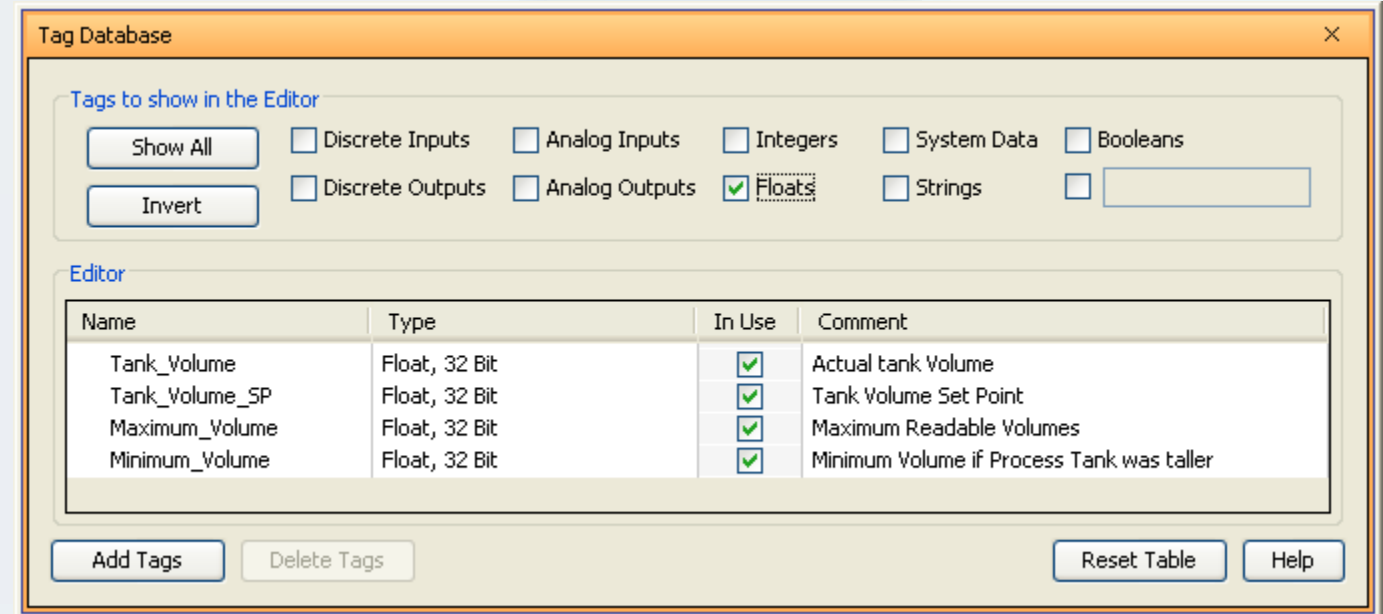
DATA TYPES:

Looking at the Tag Database, Floating Point and 32-bit Integer Data Types were assigned to the various tagnames used in the PID Loop Instruction.

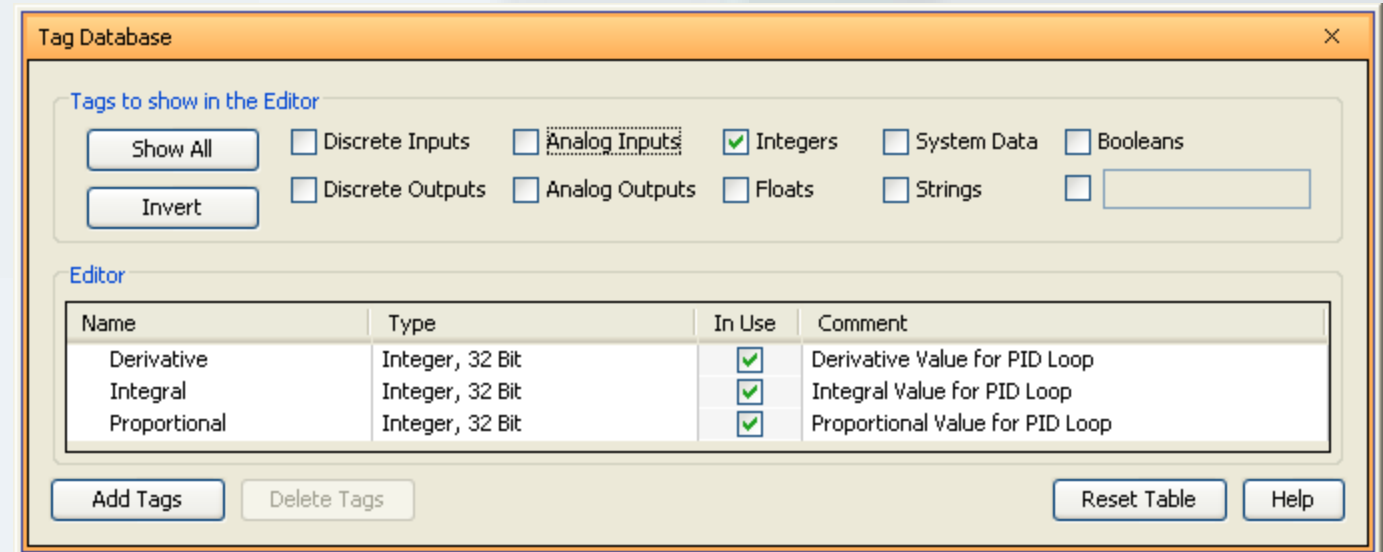
Floating Point data types were chosen for the Process Tank's volume, Set Point and minimum/maximum volume range to allow for measurements read to the hundredth of a gallon.

A 32-bit Integer data type was chosen for the Proportional, Integral and Derivative terms. The 32-bit Integer has plenty of resolution for the values that are typically used for final tuning of the PID loop.

(cont'd)



Tagname Database



Tagname Database

PID Loop Instruction – 3 of 3

The PID Loop instruction requires two rungs of control. The first is for enabling the PID execution, and the second for setting the PID to automatic mode or manual mode.

Discrete bits, controlled from push buttons on the *C-more* Touch Panel, are used to “Enable” the PID loop and also for placing the loop in “Auto” mode.

When the PID loop is not in “Auto” mode, it is in “Manual” mode. We will see how “Manual” mode is used for initial startup, and Auto-tuning the PID loop.

Productivity 3000 PAC Ladder Logic

PID Loop Instruction - Loop Name: Water Volume Control

The Enable and Manual/Auto functions are controlled by discrete bits from the C-more Touch Panel, PID_Enable and PID_Auto. Tagnames are used for the Set Point, Process Variable, Process Output, Input Range Max & Min, Proportional, Integral and Derivative. The Process Output to the 0-10 VDC analog output module is set with a range of 0 to 65.535.

		PID	
Loop Name	Water Volume Control	Output Type	Analog Out
Set Point	Tank_Volume_SP	Process Output	Pump_Speed
	4.0		18704
Process Variable	Tank_Volume	Output Range Max	65535
	3.99816	Output Range Min	0
Input Range Max	Maximum_Volume		
	6.8053		
Input Range Min	Minimum_Volume		
	-3.64968		
Gain (Proportional)	Proportional		
	25		
Reset (Integral) Time	Integral		
	5		
Rate (Derivative) Time	Derivative		
	0		

Miscellaneous Logic – 1 of 2

The process tank includes a float switch mounted through the tank's lid and is used to "kill" the power via the Power On/E-Stop circuit to prevent water overflow. Included in the ladder logic is a rung shown here to sound the audible alarm if the float switch is actuated.

A three color Stack Light tower is also used to indicate if the desired volume is OK, below desired volume, or above desired volume. Here are a couple of Math Instructions that are being used to create a .02 gallon upper and lower alarm point value used to compare between actual and set point. (cont'd)

Productivity 3000 PAC Ladder Logic

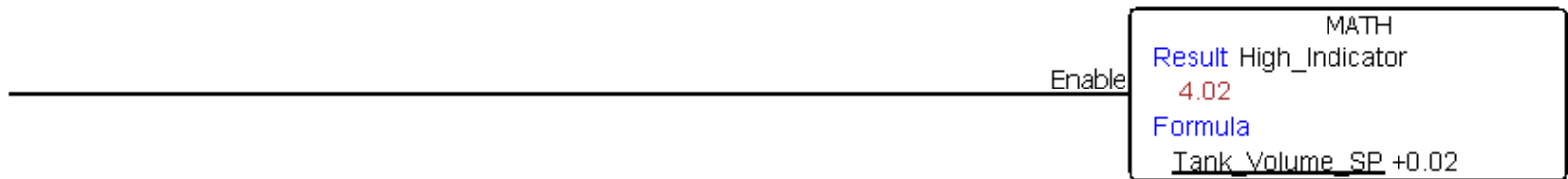
Overflow Alarm

If the float switch mounted at the top of the process tank is actuated, the 24 VDC Overflow Shutdown Relay will de-energize and the relay's N.O. contact will signal the Alarm Horn to sound, and turn the power off disabling the water pump.



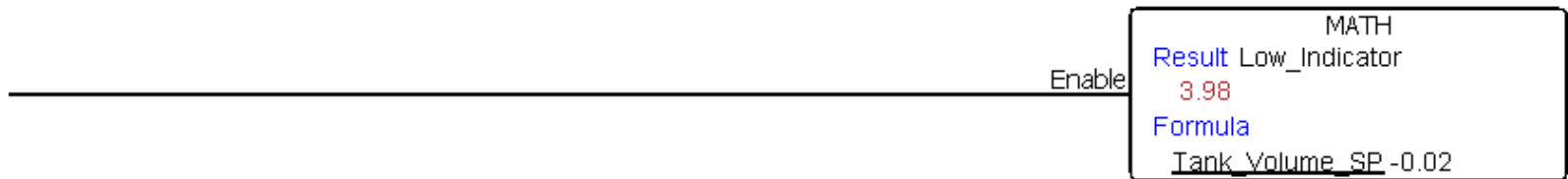
Calculate High Alarm Value

Math instruction used to calculate a fixed value of 0.02 gallons above the current Set Point to use as a comparison that will turn on the Stack Light high alarm red indicator when actual volume is greater than the High_Indicator value.



Calculate Low Alarm Value

Math instruction used to calculate a fixed value of 0.02 gallons below the current Set Point to use as a comparison that will turn on the Stack Light low alarm amber indicator when actual volume is less than the Low_Indicator value.



Miscellaneous Logic – 2 of 2

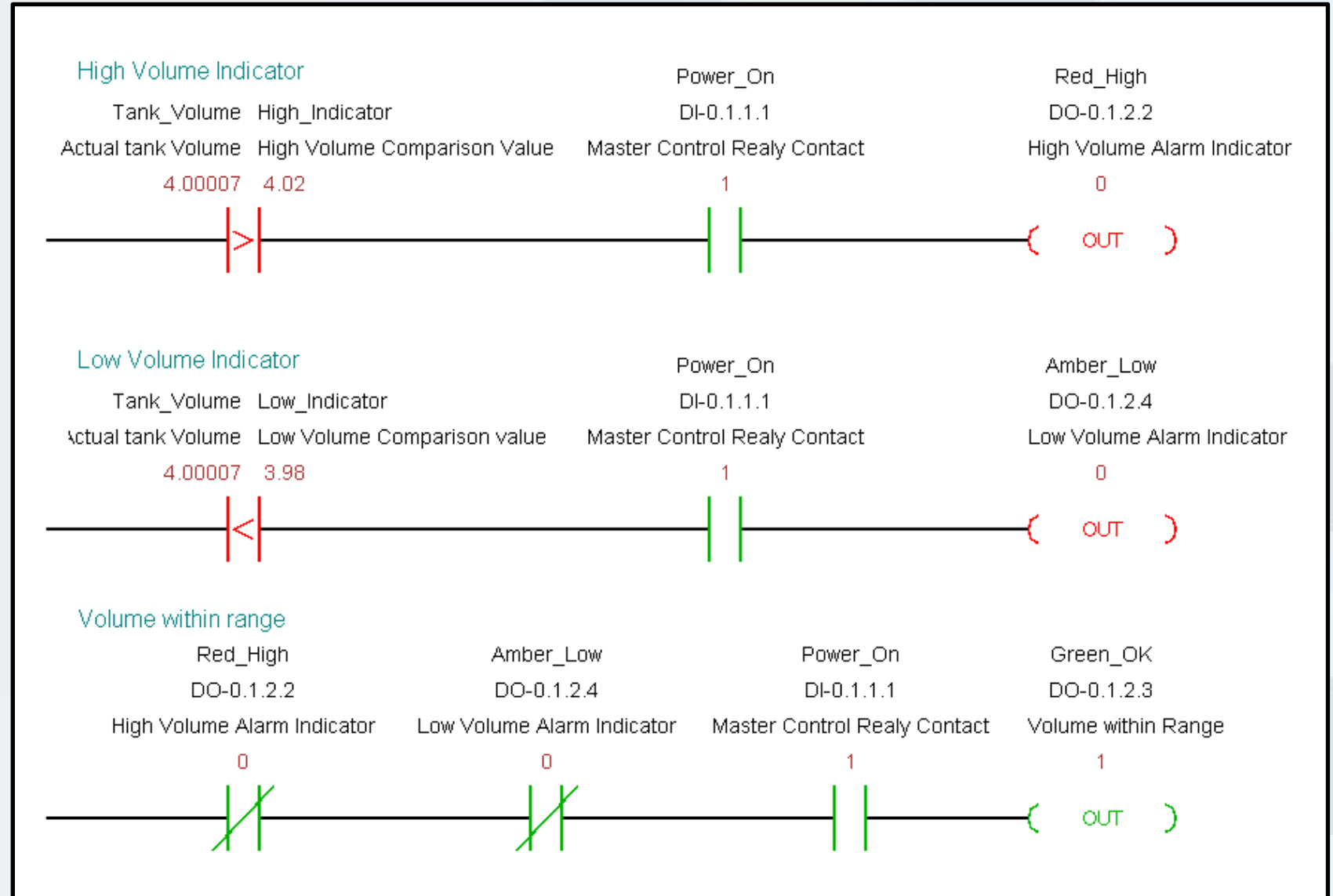
The logic to control the three various colored Stack Lights consists of three simple contact and coil rungs, and the use of the Compare Contact Instruction.

The Red indicator is lit when the actual tank volume is “greater than” the “High_Indicator” calculated value.

The Amber indicator is lit when the actual tank volume is “less than” the “Low_Indicator” calculated value.

And finally, the Green indicator is lit when the volume is within range based on the actual volume not being above or below the calculated high & low indicators.

Productivity 3000 PAC Ladder Logic



C-more PID Faceplate & Parameters Screens – 1 of 2

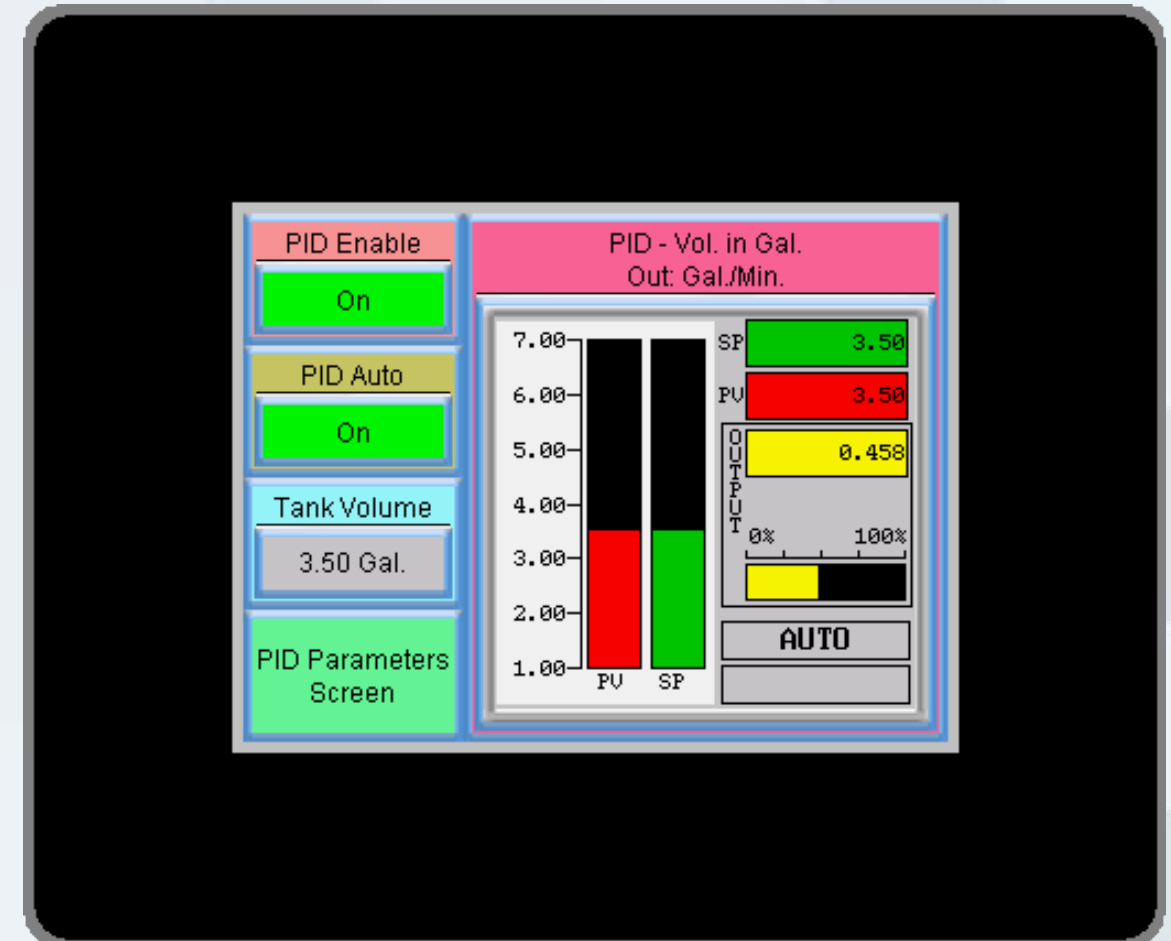
As a finishing touch to our PID Loop application, we used a *C-more* Touch Panel. The *C-more* application software has within its library a PID Faceplate Bar Meter that works perfectly with our PID application. The use of the faceplate makes PID setup, control, monitoring and tuning a breeze.

The PID Faceplate incorporates bar graphs for the Set Point, Process Variable and Process Output. It also displays status of operation mode (auto/manual) and alarm monitoring.

Along with the PID faceplate, we have used the *C-more* to add push buttons to enable the PID loop and selecting the auto or manual mode. A Numeric Entry object from the *C-more* object list is used to enter the Set Point for the process tank's desired volume.

A Screen Change push button is used to get us to the PID Parameters screen. (cont'd)

C-more Touch Panel



C-more PID Faceplate & Parameters Screens – 2 of 2

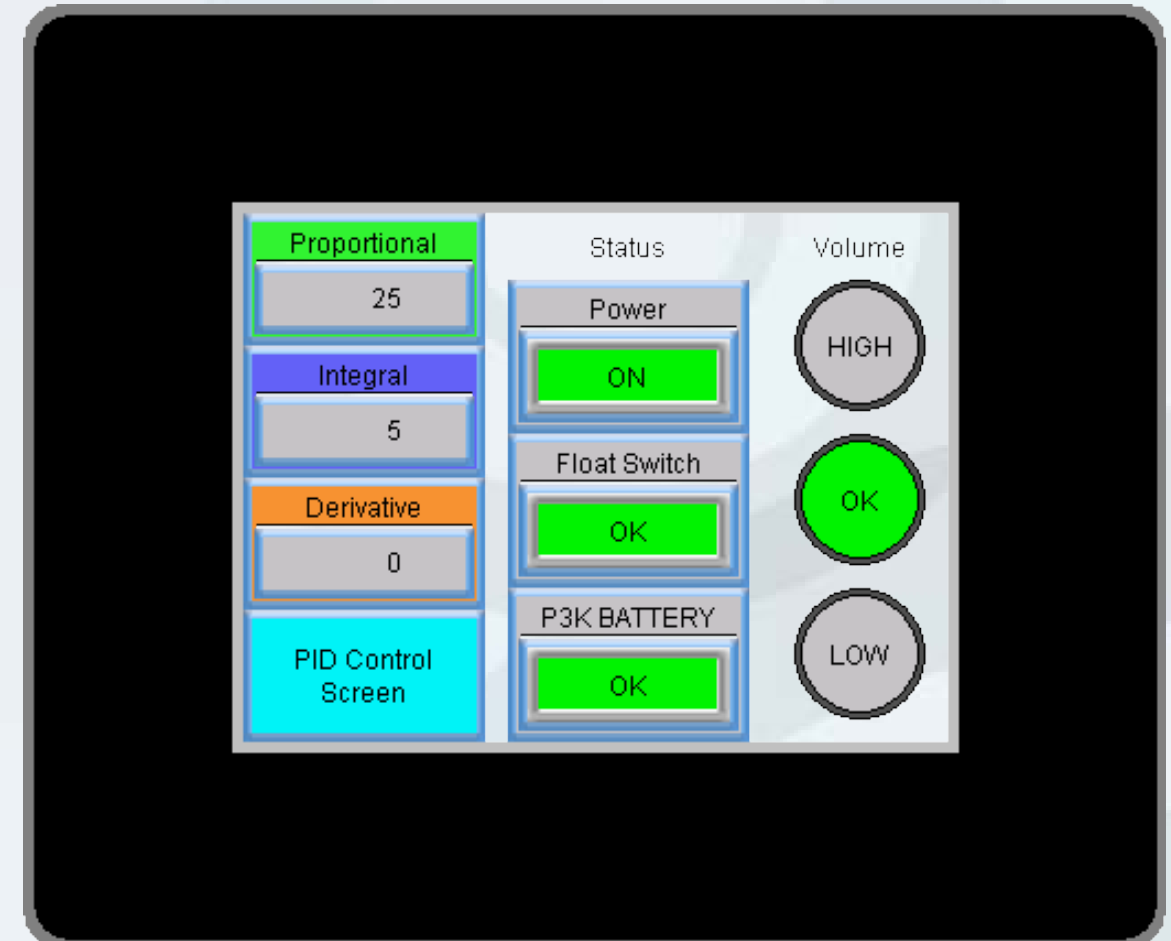
A second *C-more* screen is available to allow manually adjusting the Proportional, Integral and Derivative term values as required.

This screen also includes status indicators (Indicator Lights) for the main control power, overflow float switch condition, and the health of the Productivity 3000 optional battery used to retain the Time and Date, and retentive tagname values.

The Stack Light tower indicators used to show the process tank's volume condition are duplicated on this screen. This is ideal if remote access to the *C-more* were done over the internet.

Finally a Screen Change push button is available on this screen to get us back to the main PID Loop monitoring screen.

C-more Touch Panel



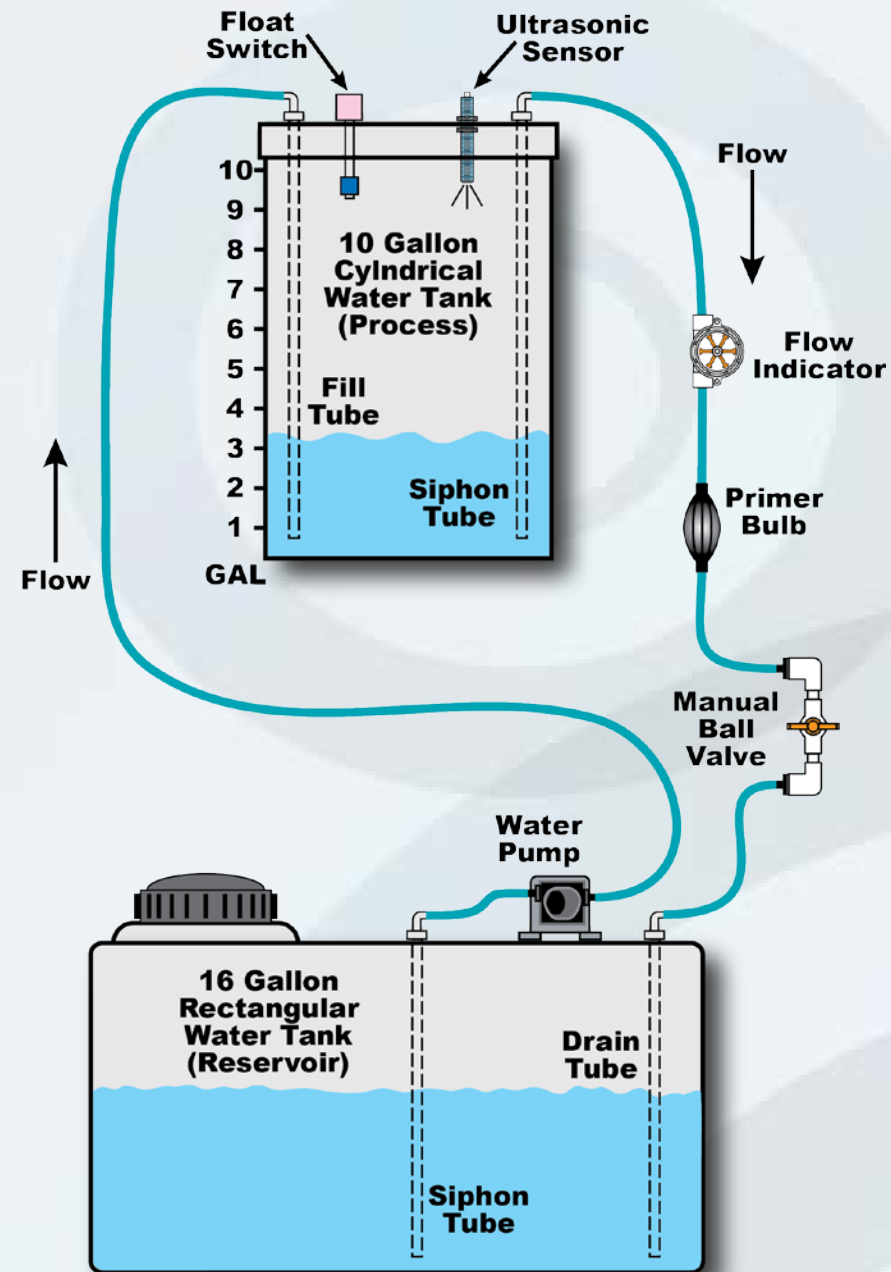
PID Loop Tuning – Manual – 1 of 4

Once we have all of the equipment in place, wired, and checked out, we can next start the process, monitor it with the built-in PID Tuning feature, and step through manual tuning to test functionality and check response.

Start with applying power to the equipment and press the Power On push button. Via the *C-more* touch Panel, enter a value for P, I & D as described on the next slide, enter a volume such as 3.00 gallons, enable the PID, and enable Auto mode. The pump should run at maximum output filling the process tank from the reservoir tank.

Our purpose at this point is to get enough water into the process tank so that the level is above the bottom of the siphon tube located toward the bottom of the process tank.

(cont'd)



PID Loop Tuning – Manual – 2 of 4

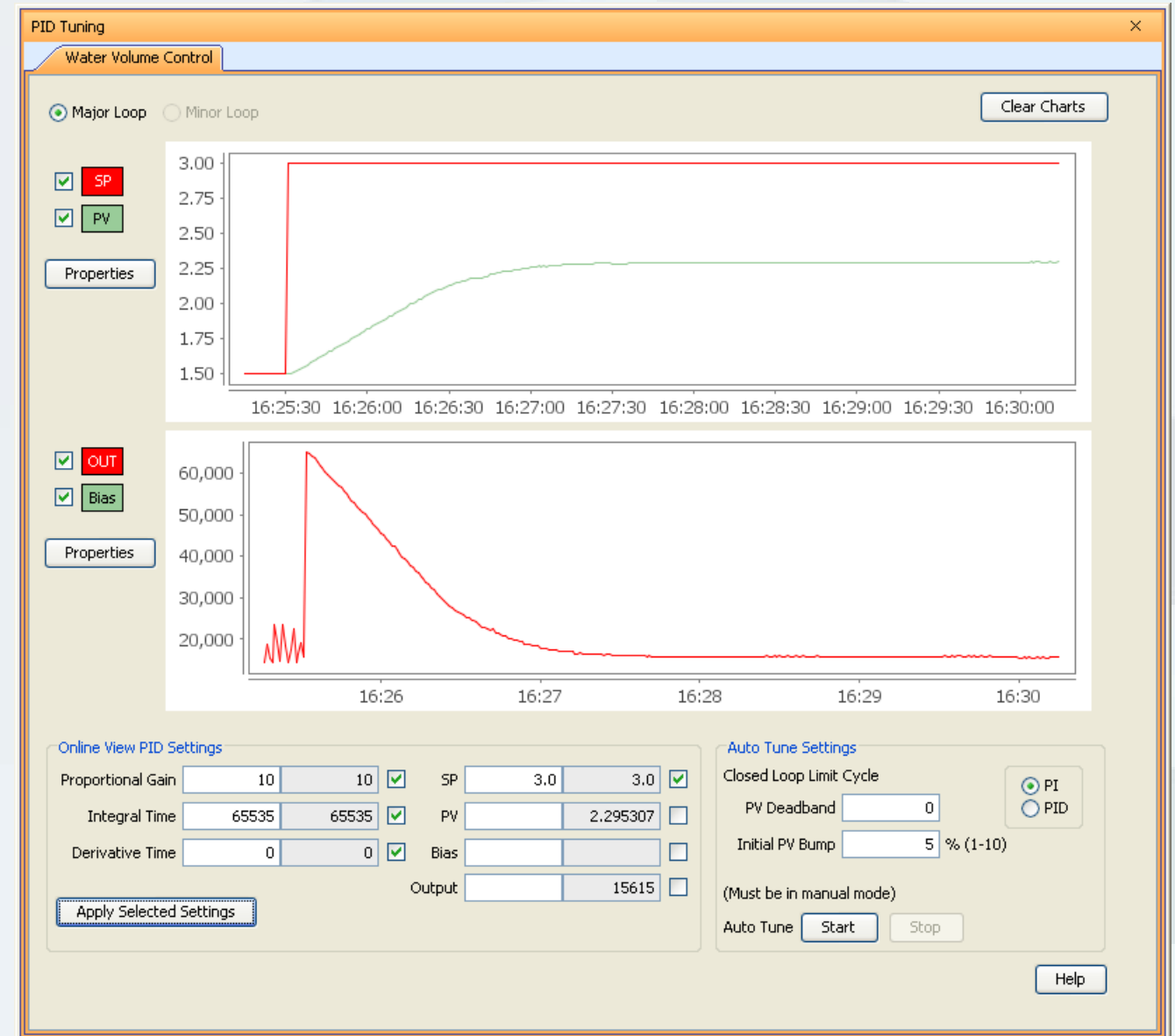
Under safe conditions, and being ready to hit the Emergency Stop push button, it is usually a good idea to start with a Proportional value of '10', disable the Integral reset by entering the maximum value of '65,535', and also disable the Derivative term by entering '0'.

As can be seen with the PID Tuning window, the volume has trouble reaching our Set Point, mainly because a Proportional gain of '10', along with no Integral reset time, just isn't enough for our particular application.

But once we have a couple of gallons in the process tank, we can open the ball valve and use the primer squeeze bulb to start a drainage flow.

This will get us ready for the next step, which is making manual changes to our P, I and D term values to have the pump increase the level to obtain the desired volume. (cont'd)

PID Tuning Monitoring

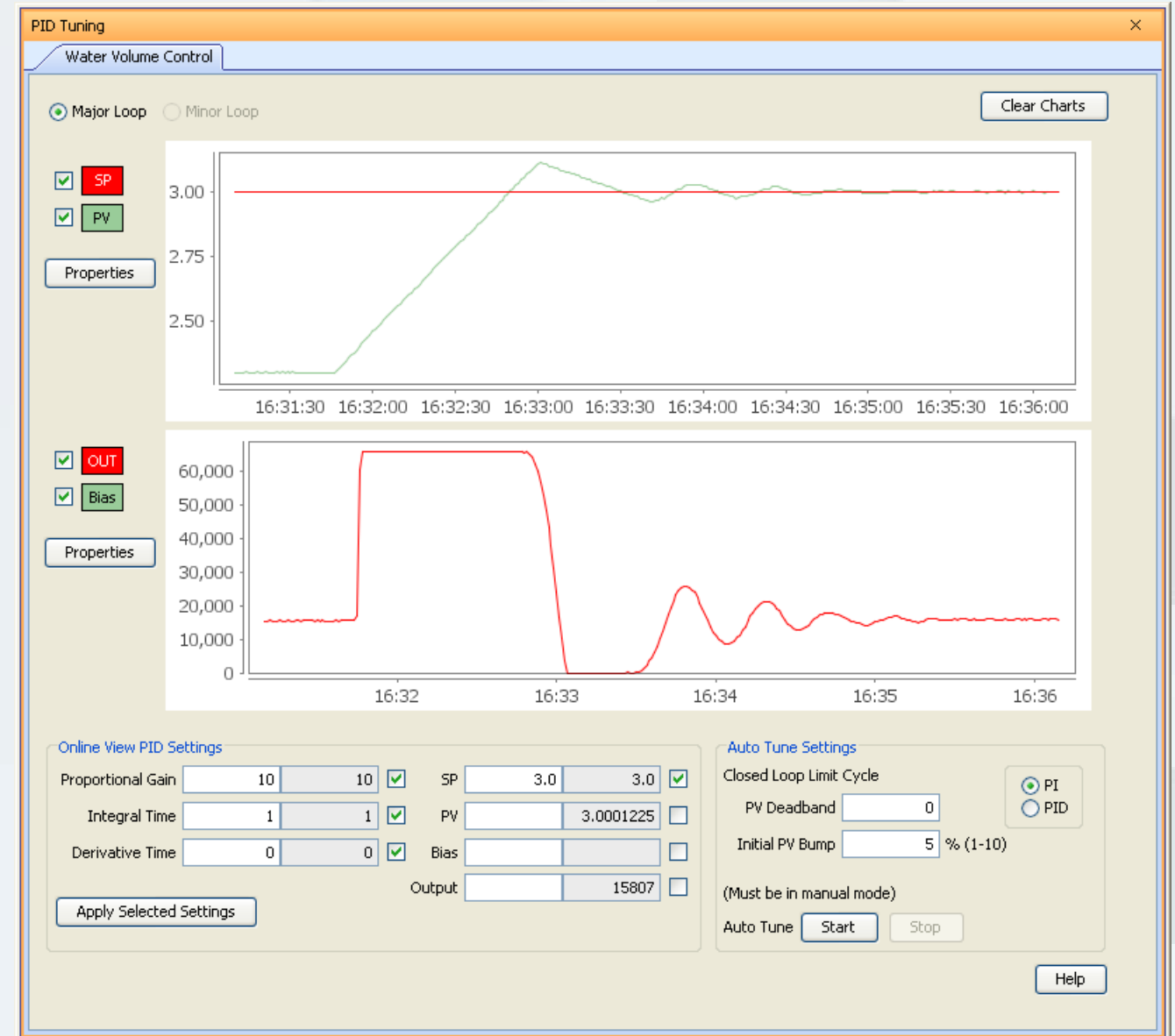


PID Loop Tuning – Manual – 3 of 4

Without changing our Proportional gain at this time, we change the Integral reset time to '1' which allows for a faster correction of the accumulation of 'past' errors, as we learned earlier in this tutorial. As seen with the PID Tuning window, the pump is driven at full output, bringing the volume up to our Set Point value, with some overshoot. After oscillating around the Set Point for a period of time, the volume settles down, but the time it takes to do so is very slow.

Our next step in manually tuning the PID Loop will be to increase the Proportional gain to obtain a faster response, enter a new Set Point value, monitor the response using the PID Tuning window, and make adjustments to the Integral reset time value to reduce oscillations, also called hunting. (cont'd)

PID Tuning Monitoring



PID Loop Tuning – Manual – 4 of 4

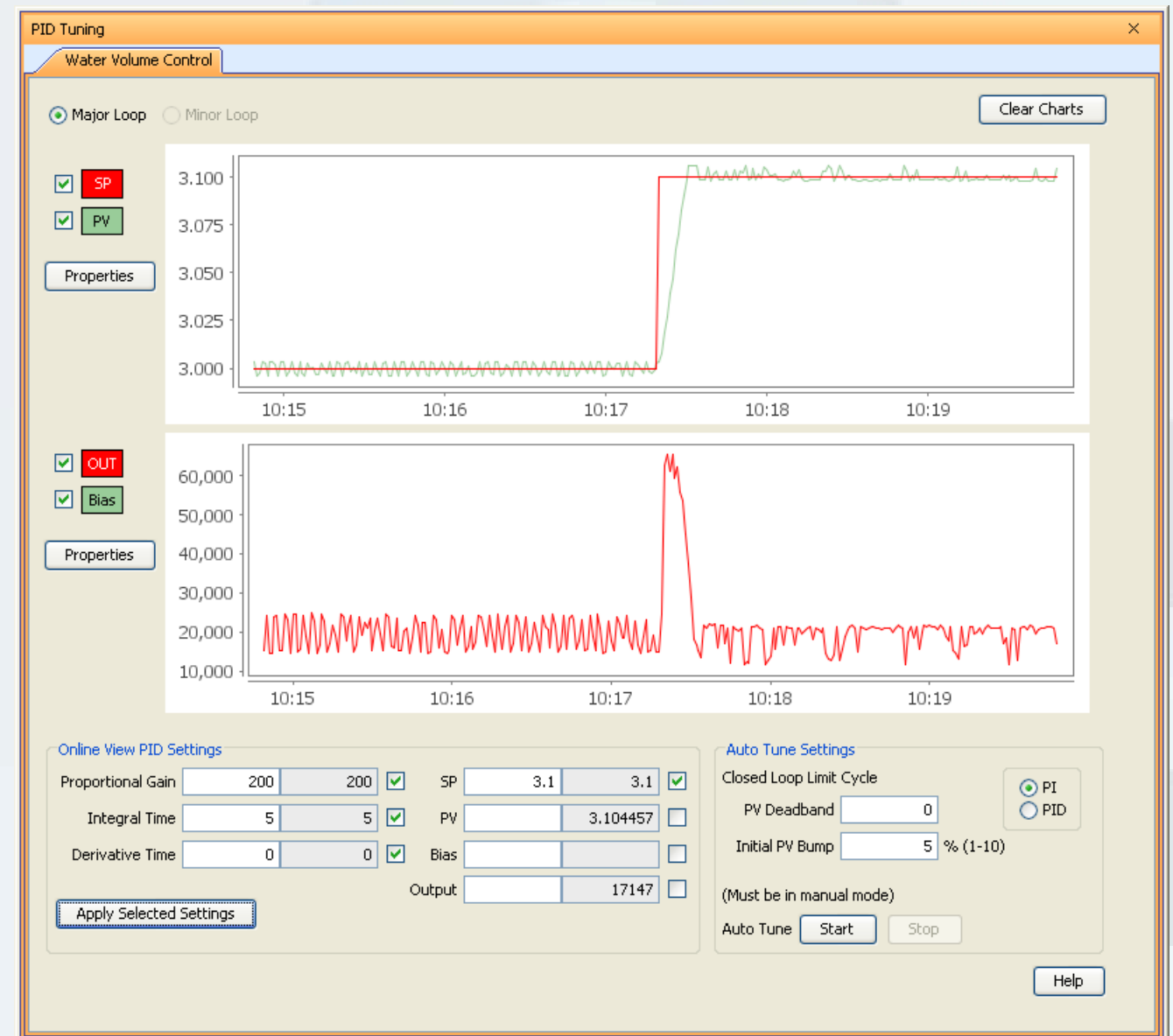
The manually tuning process takes some trial and error in making changes to the various Proportional and Integral term values, changing the Set Point, waiting for the results while monitoring the response, and repeating these steps. Eventually you should see improvement with the PV getting to the SP with little overshoot and no oscillations.

Our goal is to obtain optimal performance in the PID Loop response, and seen here we were able to obtain our best results with a Proportional gain of '200' and an Integral reset time value of '5' seconds.

Because our process is a slow responding type application, we do not require any Derivative time. In fact, in our application, adding any Derivative actually causes the PID loop to become unstable.

Next: The built-in Auto-Tuning feature.

PID Tuning Monitoring



PID Loop Tuning – Auto – 1 of 2

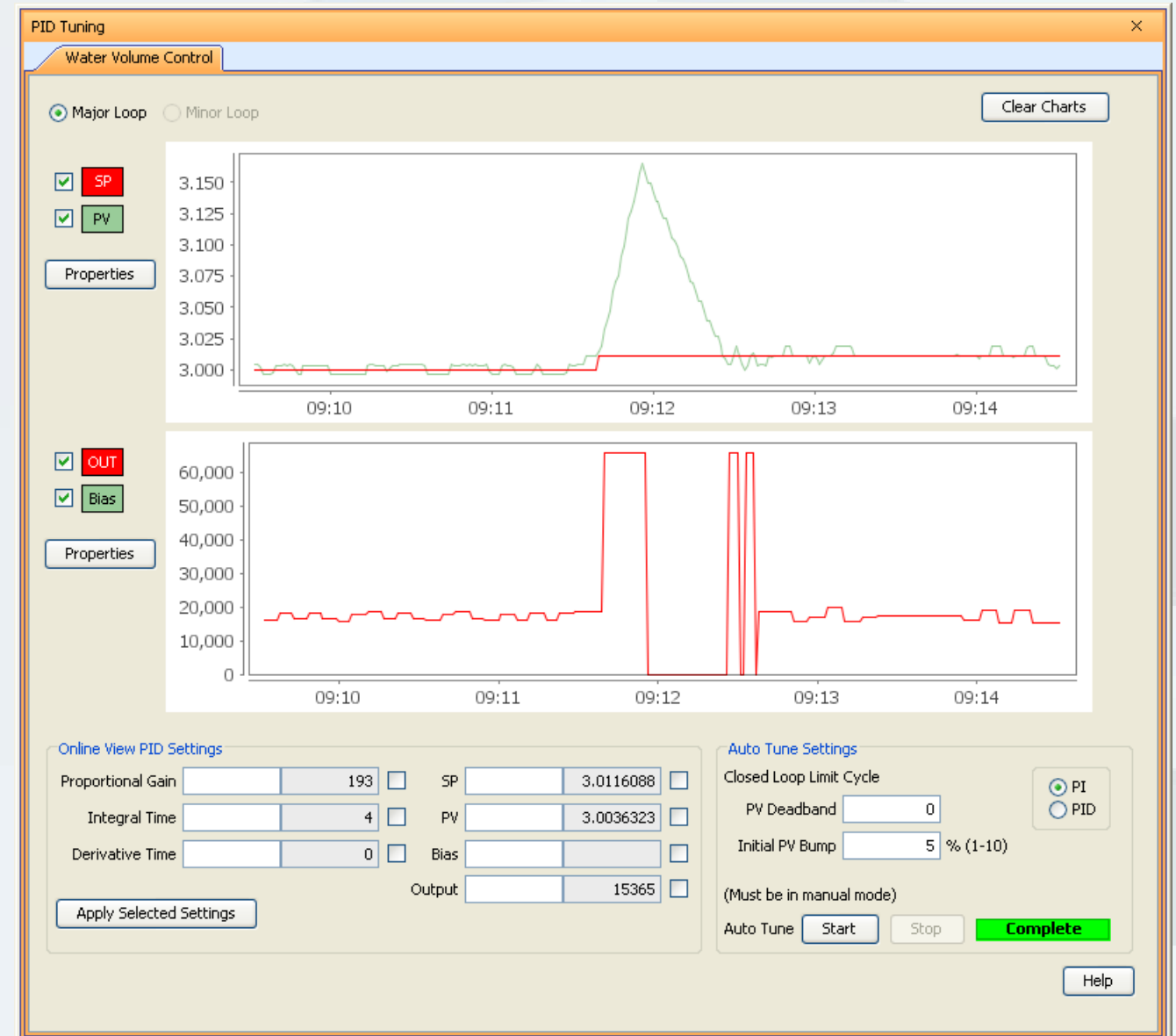
To use the built-in Auto Tune feature, switch to Manual Mode and place the PID loop in a steady state by manually adjusting the Process Output value. We used an Initial PV Bump of 5% and leave the PV Deadband at '0'. Again our process did not require a Derivative value, so only the PI radio button was checked. Now click on the Auto Tune 'Start' button in the PID Tuning window.

The current Process Variable will be copied to the Set Point value, in our example this became a volume of 3.0116088 gallons.

The Process Output will go to the upper limit (65,535) and drive the pump full on to increase the volume in the tank.

Once the Process Variable reaches the Initial PV Bump, in our case 5%, or approximately 3.15 gallons, the Process Output will go to the lower limit (0). (cont'd)

PID Tuning Monitoring



PID Loop Tuning – Auto – 2 of 2

When the Process Variable falls to the Set Point, the Output is then driven to the Output upper limit.

When the Process Variable rises to the Set Point, the Output is driven to the Output lower unit again.

The Output cycling pattern continues for three full cycles, and when completed, the Auto Tune routine calculates the new values for the P, I & D terms and writes them to the appropriate Tagnames.

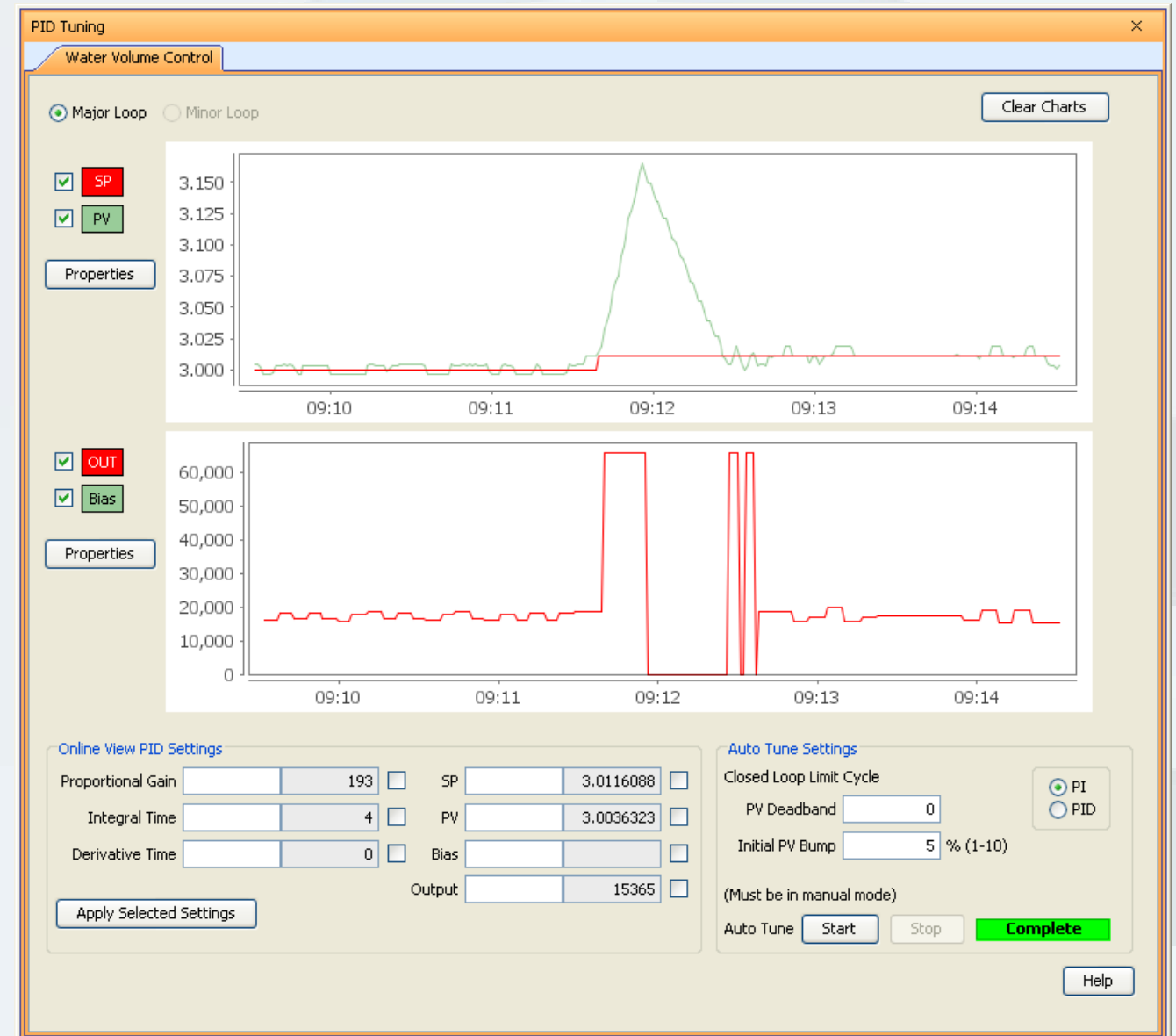
Proportional Gain = 193

Integral Time = 4

Derivative Time = 0

These values are almost identical to the values that were determined by way of our trial and error manual tuning effort.

PID Tuning Monitoring



PID Loop Application Demonstration – 1 of 3

SUMMARY:

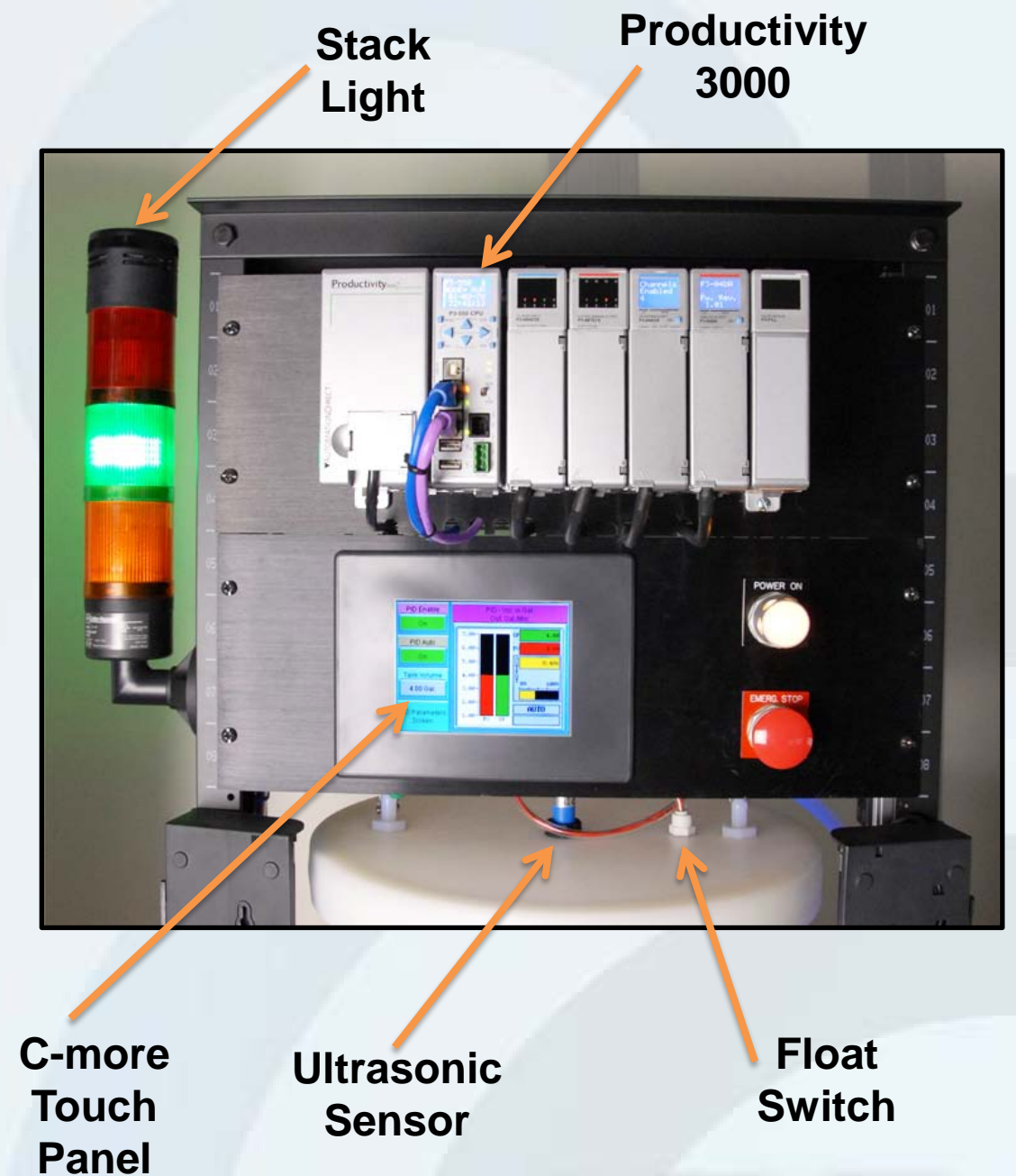
The PID Loop application example in this tutorial uses the Productivity 3000 Programmable Automation Controller to maintain a settable volume of liquid in a cylindrical tank.

An ultrasonic sensor that outputs a 0-10 VDC analog signal to an analog input module, plugged into the base of our Productivity 3000 system, measures the level in the process tank, and with Math and Scaling programming instructions, a volume is calculated.

A diaphragm motor/pump, whose rate, via a DC motor controller, is controlled by a 0-10 VDC analog output module, also plugged into the base of our Productivity 3000 system, replenishes the liquid in the process tank as it is used.

The tank volume Set Point for our PID loop is entered via a *C-more* Touch Panel. A PID Faceplate Bar Graph from the *C-more's* software object list is used to monitor the Process Variable and Process Output.

(cont'd)



PID Loop Application Demonstration – 2 of 3

SUMMARY (cont'd):

A ball valve is located in the drain line that allows the rate at which the liquid drains back to the 16 gallon reservoir to be adjusted. There is also a flow indicator to show a visualization of how fast the 10 gallon process tank is draining.

A primer squeeze bulb is used to start the drainage from a siphon tube located in the process tank.

A Stack Light with three different colored indicators display if the actual tank volume is above, below or OK as compared to the desired volume.

See the accompanying LEARN video to see the PID Loop application in action!

(cont'd)



PID Loop Application Demonstration – 3 of 3

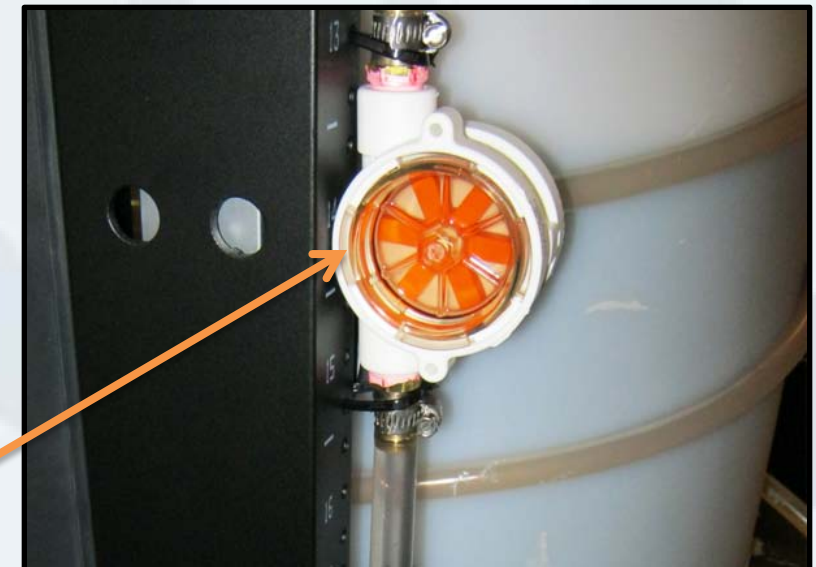
SUMMARY (cont'd):

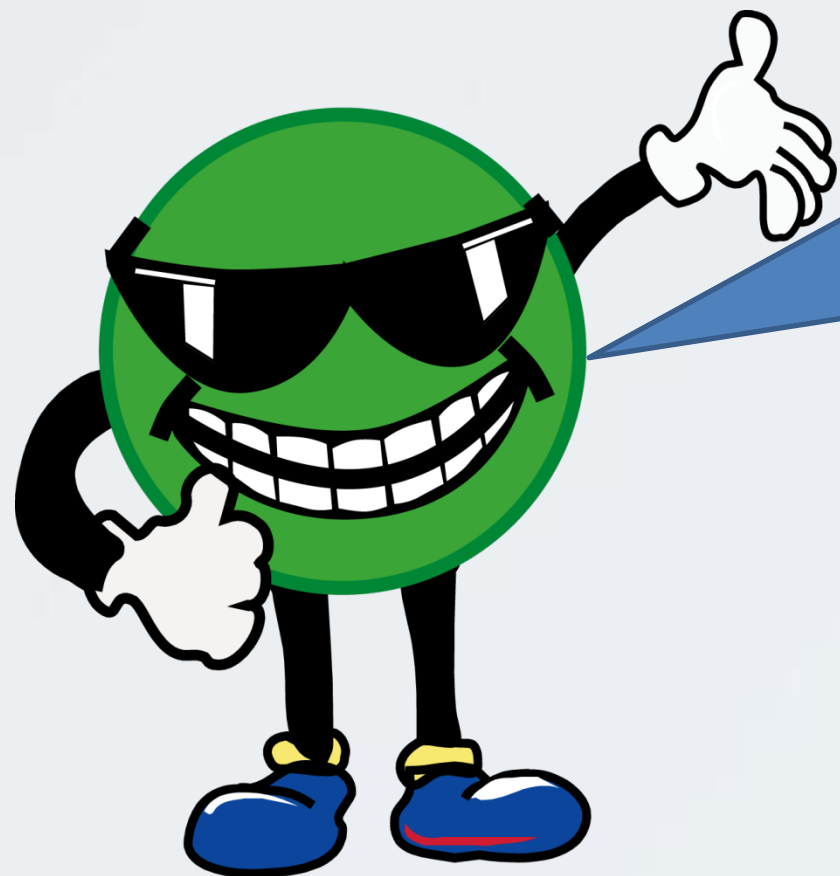
As seen here, a paddle wheel type flow indicator was located in the drain line to give a visual indication of the drainage rate.

You may ask where an application to control the volume in a tank may be used?

Picture a situation where we are mixing a solution, and we need to maintain a continuous concentration, but the mixed solution is being used at an ongoing random rate. Therefore we would need to maintain the overall volume while measuring how much solution is used at any given instant, and adjust how much additive needs to be replaced accordingly. This would be another good task for the Math Instruction available in the Productivity 3000 Programmable Automation Controller!

Flow
Indicator





That's it for now! I don't have anything else to say about PID loops, but hopefully what was covered in this presentation and the accompanying LEARN video will get you on your way the next time you are up against the wall on an application requiring a PID loop, and you have the wisdom to use the Productivity 3000 to make the challenge much easier to accomplish!

Available videos in the Productivity 3000 – Be More Productive – Programming (PID Loop) series:

Title	VID Number
Part 1 of 11 – What is a PID Loop and what does it do?	L-PC-P3K-005-1
Part 2 of 11 – How does a PID Loop work?	L-PC-P3K-005-2
Part 3 of 11 – Application Example and Hardware Explained.	L-PC-P3K-005-3
Part 4 of 11 – Programming – Hardware Configuration, Tagnames, and Calculations.	L-PC-P3K-005-4
Part 5 of 11 – Programming – More Math Calculations.	L-PC-P3K-005-5
Part 6 of 11 – Programming – Scaling.	L-PC-P3K-005-6
Part 7 of 11 – Programming – PID Loop.	L-PC-P3K-005-7
Part 8 of 11 – Programming – Miscellaneous Logic.	L-PC-P3K-005-8
Part 9 of 11 – Programming – C-more Touch Panel.	L-PC-P3K-005-9
Part 10 of 11 – PID Tuning.	L-PC-P3K-005-10
Part 11 of 11 – PID Loop Application Demonstration.	L-PC-P3K-005-11

Please note.

Learn.AutomationDirect.com is an online streaming tutorial site offering training and information on a wide range of practical automation products.

THE LEARN.AUTOMATIONDIRECT.COM WEBSITE, AND THE TRAINING AND INFORMATION PROVIDED IN CONNECTION THEREWITH, IS SUPPLIED "AS IS". These video presentations and other documents are provided by our associates to assist others in learning the products we sell and service. We make no representation, warranty or guaranty, whether expressed, implied or statutory, regarding the LEARN.AUTOMATIONDIRECT.COM website on the training, information and the content, including without limitation, the implied warranties of merchantability or fitness for a particular purpose, and any representation, warranty or guaranty that the foregoing will be accurate, complete, uninterrupted error free or non-infringing, is suitable for your particular application, nor do we assume any responsibility for the use of this information in your application.

User suggestions, corrections and feedback are not only welcomed, but are essential to the maintenance of current content and the creation of new content. If there is a training idea, or correction to an existing presentation you would like us to consider, please complete and submit the suggestion form that is shown as a "Suggestions" link at the bottom of every page.

Thank you!