

PID Loop

Table of contents

PID Loop 3

 Introduction To PID 3

 PID Setup 5

 PID Monitor 12

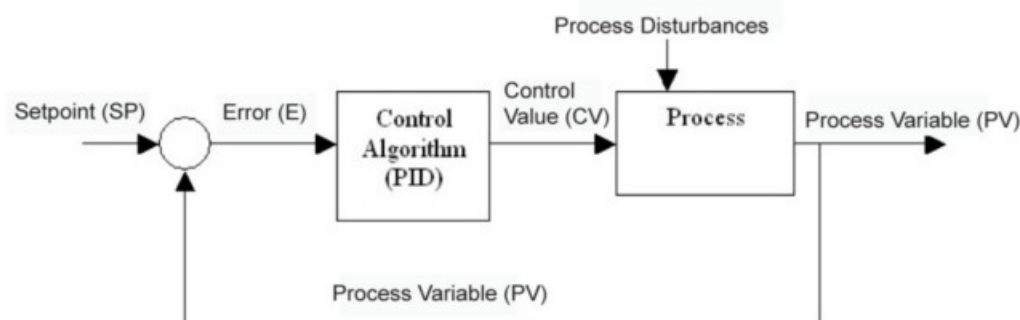
 PID Loop Auto Tuning 15

PID Loop

Introduction To PID

Industrial Manufacturing Processes involve many variables such as temperature, pressure, flow, etc. It is important to control these variables for proper operation of the process. There are several methods to control process variables. PID is one of the most popular control algorithms used in the industry. PID, as applied to Industrial Process Controls, stands for Proportional, Integral and Derivative control algorithm. The algorithm computes control action by using a mathematical equation which contains Proportional, Integral (Reset) and Derivative (Rate) terms. With proper choices of P, I, and D terms, a user can maintain a process value very close to the Setpoint. In addition, if the Setpoint or the process dynamics changes, the PID algorithm can bring the process back under control quickly. Selection of appropriate P, I and D coefficients is critical to the proper operation of the PID control.

A block diagram of a generic process control is given below:



As shown in the figure, the user sets a target or Setpoint for the process. The system compares the actual Process Variable against the Setpoint and generates an Error value. The PID algorithm uses this error and computes a Control Variable as a function of the error. The computation function contains P, I, and D terms with user defined coefficients. The PID algorithm's goal is to minimize the error. If the Setpoint changes or the process is disturbed (resulting in a change in the Process Variable), a new error value is generated which results in a new Control Variable that should bring the Process Variable closer to the Setpoint.

PID Terminology

Before we discuss more of the details involved with the PID Loop, you should have an understanding of some of the terms used in PID.

Manufacturing Process - A process that transforms a material's properties. The transformation may involve physical or chemical changes in the material. Examples of processes are: Steam Generation, Air conditioning, Milk Pasteurization, Oil refinement, etc.

Process Variable - Materials that have physical measurable properties, such as temperature, volume, viscosity, pressure, etc. A Process variable is a measurable physical property that we want to control. For example, in the air conditioning of a building, we want to control temperature, and therefore temperature is the Process Variable.

Setpoint Value - The target or desired value of the Process Variable. The purpose of PID loop is to maintain the Process Variable as close to the Setpoint as possible.

Control Variable - The Control Variable is calculated by a control algorithm. It depends on the error and PID coefficients. **(see next section for the equations used in the**

computations).

Error - Error equals the algebraic difference between the process variable and the setpoint. Magnitude and variation of the error depends on the process dynamics as well as on the PID coefficients. A well designed system will keep the error to a minimum value.

External Disturbance - Something that changes the equilibrium of the process. This results in a change in the control action to bring the process back into range. For example, in an air conditioned building, open doors and rainstorms are all changes that can affect the temperature.

PID on EZ Series PLC

EZ Series PLC products support up to 8 PID loops. For each loop the user defines several parameters (such as Setpoint, Proportional, Integral (Reset) and Derivative (Rate) Gains, Limits, etc. **(further discussed in the next section)**). You can change most of these parameters at run time using ladder logic by using the EZ Series PLC Editor software in online mode.

Note: To use the PID Loop feature, you will need firmware revision B.0 or later.

To view the firmware revision of the EZ Series PLC, go to the EZPLC Menu and select INFORMATION.

PID Algorithms used in EZ Series PLC

The EZ Series PLC uses the following algorithms for PID computations:

Let ,

SP_n = Setpoint at sample instance n
 PV_n = Process Variable at sample instance n
 CV_n = Control Variable at sample instance n
 K_p = Gain, Proportional term
 T_i = Reset (integral) time in seconds
 T_d = Derivative or React time, in seconds
 T_s = Sample time in seconds
 E_n = Error at sample instance n
 CV_0 = Control Variable offset

Position Algorithm

$$CV_n = K_p * [E_n + (T_s/T_i) * \sum_{i=0}^n E_i + (T_d/T_s) * (E_n - E_{n-1})] + CV_0$$

Velocity Algorithm

$$CV_n = K_p * [E_n + (T_s/T_i) * \sum_{i=0}^n E_i + (T_d/T_s) * (PV_n - PV_{n-1})] + CV_0$$

Note: There are options in the setup that will modify the CV computations.

For example, the user can choose to use PV Square root instead of PV in error computations.

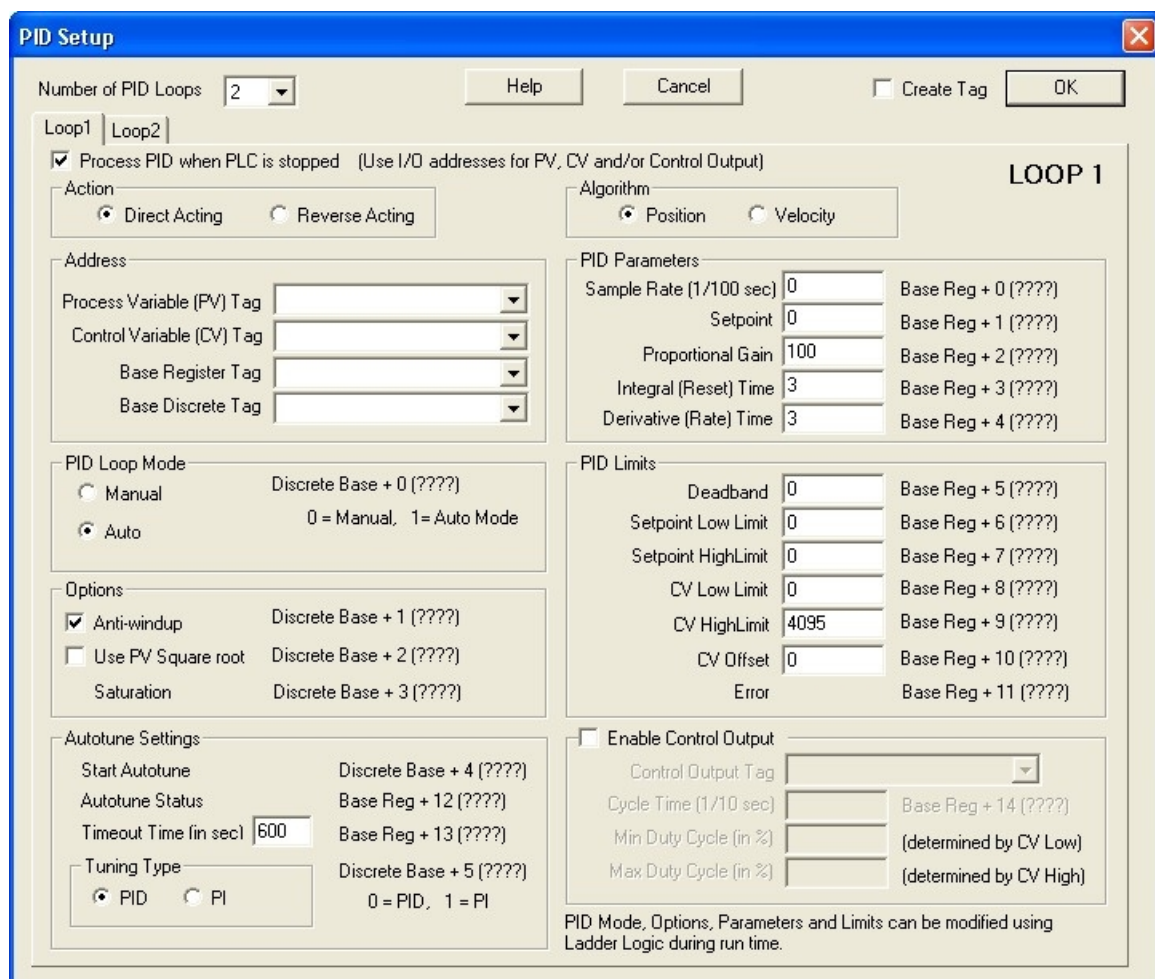
Please see the PID setup where these options are discussed.

PID Setup

The following section will explain how to setup a PID loop using your EZPLC Editor software. To access the PID Setup, perform the following steps: Go to the Setup Menu and select PID. The following dialog box will appear (If you have already defined one or more loops, the image below will be different).



Use the drop-down arrow to select the Number of PID Loops you would like to use (you can select up to 8 PID Loops). As soon as you select a number of loops other than 0, the following dialog box will appear:



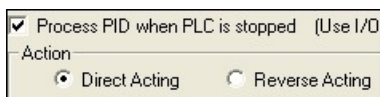
The Dialog box above allows you to define all your PID Parameters. It will show as many tabs as the number of PID loops selected. The tabs are labeled Loop1, Loop2, and so on. Each PID loop requires a contiguous block of 32 Registers and a contiguous block of 8 discrete for storing parameters and status. The blocks start at user-specified starting base addresses. In addition to the start-of-block of addresses, following tags are required: Process Variable, Control Variable and, optionally, Control Output.

The user defines the Base (or Starting) address/tag of the Register Block. The EZ Series PLC then maps the next 31 registers automatically, making a total of 32 registers per block. Out of the block of 32 registers, 15 are used currently, and the rest are reserved for future use. Similarly, the user defines a Base (or Starting) address/tag for the

Discrete Block. Then the EZ Series PLC will map the next 7 addresses automatically, making a total of 8 discrete in the block.

The dialog box shows which register of the block will store what for your ready reference. For example, the Sample Rate is stored in the first register of the Block (Base+0), while Deadband is in Base + 5 register. Since you know the addresses of all parameters, you can define these parameters in this dialog box, and/or dynamically define/modify these using ladder logic during runtime. The buttons and fields that appear in the PID Setup dialog box are explained below.

Process PID when PLC is Stopped - When PLC is stopped (not in Run Mode), it does NOT process ladder logic or Update I/O. However in some cases, it may be desirable to continue the PID loop even when the PLC is stopped. Use this check box to indicate that the PID should be processed when the PLC is stopped. The default is to continue PID processing.



The screenshot shows a dialog box with a checked checkbox labeled "Process PID when PLC is stopped (Use I/O)". Below it, under the "Action" section, there are two radio buttons: "Direct Acting" (which is selected) and "Reverse Acting".

***Note:** If you want the PID to be processed with PLC stopped, make sure to use physical addresses for the PV, and CV/Control Output tags.

The reason is that when PLC is stopped no ladder executes, and ONLY PID related I/O would be updated if the "Process PID when PLC is Stopped" is checked.

Action - You must select either Direct Acting or Reverse Acting. In a Direct Acting loop, the Control Variable and Process Variable follow the same direction i.e. Increase in Control Variable increases the Process Variable and vice versa. For example in a heating application, the more power through a heater (CV) increases the temperature (PV). The Reverse Acting loop, the CV and PV move in opposite direction. SO an increase in CV decreases the PV. For example, in an air conditioning or cooling application, more power is applied to reduce the temperature.

EZ Series PLC computes error term, based on this choice, as follows:

$$E_n = PV_n - SP_n \text{ for Direct Acting}$$

$$E_n = SP_n - PV_n \text{ for Reverse Acting}$$

Algorithm (Position or Velocity) - EZ Series PLC supports two PID algorithms, know as Position and Velocity algorithms. Select whether you would like to use a Position math equation or a Velocity math equation.



The screenshot shows a dialog box with a section titled "Algorithm". It contains two radio buttons: "Position" (which is selected) and "Velocity".

Process Variable (PV) Tag - Use the drop-down arrow or enter a tag address where you would like the Process Variable to be stored. You can use R or IR register types. If you use an IR type tag, then you are reading the Process Variable directly from an Input Module. If use an R-type after some scaling) using logic to the R-type register so that PID computations can use the PV.



The screenshot shows the "Address" section of a dialog box. It contains four drop-down menus: "Process Variable (PV) Tag" (set to IR1), "Control Variable (CV) Tag" (set to OR1), "Base Register Tag" (set to R1), and "Base Discrete Tag" (set to S1).

***Note:** If you would like PID to run while the PLC is stopped, you should choose an IR type tag so that the PV is updated with the actual value.

Control Variable (CV) Tag - Use the drop-down arrow or enter a tag address where you would like the Control Variable to be stored. The CV tag has the flexibility of using R or OR registers, If you use OR, then EZ Series PLC writes the CV directly to an Output Module. If you use the R type for CV tag, you will have to move the actual CV (possibly after some scaling) using ladder to an output module for control.

***Note:** If you would like to PID to run when PLC is stopped, please use OR type tag for CV so that it can be updated, unless you are using Control Output.

Base Register Tag - Base Register Tag/Address defines the starting address of a Contiguous Block of ??? registers that are used to store PID Parameters and Status information. Please see the dialog box to find the addresses of desired parameter within the block.

Base Discrete Tag - Base Discrete Tag/Address defines the starting address of a Contiguous Block of ??? registers that are used to store PID Parameters and Status information.

PID Loop Mode - In Auto mode, the PID Loop calculates a new Control Variable value every sample period. In Manual mode, the Control Variable is controlled by user manually. The manual mode may be used for manual control of process. PID Monitor dialog box (**EZPLC>PID Monitor**) can be used to modify Control Variable in manual mode.

PID Loop Mode	
☐ Manual	Discrete Base + 0 (S1)
☒ Auto	0 = Manual, 1 = Auto Mode

When the mode is switched from manual to auto, the integral term of the PID equation is set to the control value. This provides bumpless transfer from manual to auto.

Anti-Windup - This option inhibits integration when the control value is saturated. It controls the integral term of the PID equation when the control value is saturated. If Anti-Windup is selected, the integral term is not included when the output is saturated and the sign of the Error will cause the integral term to drive the output further into saturation. This help loops to come back into equilibrium sooner.

Options	
☒ Anti-windup	Discrete Base + 1 (S2)
☐ Use PV Square root	Discrete Base + 2 (S3)
Saturation	Discrete Base + 3 (S4)

Use PV Square Root - If this option is selected, Square root of PV is used instead of PV in error computation.

Saturation - This line is for information only. This line shows the address of the discrete bit that would be set if the Control Variable is saturated (i.e. if the Control Variable is either 0 or 4095). You may use this in ladder logic to monitor the saturation of control variable.

Autotune Setup

The EZ Series PLC can autotune PID loops, i.e. it can estimate the values for the Proportional Gain, Integral (Reset) time, and Derivative (Rate) time for PID loop. The dialog box allows you to setup the loop for autotune. EZ Series PLC uses Ziegler-Nichols method to estimate the PID parameters. Following are the setup parameters for Autotune:

Start Autotune - Shown on the dialog box for information only.

The Start Autotune discrete is at Discrete Base+4. EZ Series PLC initiates autotuning of a loop when this bit transitions from 0 to 1. Autotuning of the loop is started regardless of the selected "PID Loop Mode" of the loop. Once Autotune is started, you can stop it by making setting this bit to 0.

Autotune Settings	
Start Autotune	Discrete Base + 4 (S5)
Autotune Status	Base Reg + 12 (R13)
Timeout Time (in sec)	Base Reg + 13 (R14)
Tuning Type	Discrete Base + 5 (S6)
☒ PID ☐ PI	0 = PID, 1 = PI

Note: Autotune is performed by EZ Series PLC observing open loop response to a step change in the control value. Before starting autotune, the process should be in a steady state. During Autotune, watch the process variable closely for it to be within the safe limits.

Autotune Status: Shown on the dialog box for information only. During Autotune, EZ Series PLC reports the status of Autotune in the register.

Register Value	Description
0	Tuning in progress
1	Tuning Done
2	User canceled tuning
3	Control Value could not be incremented
4	The tuning algorithm failed to fit the curve
5	Division by zero error
6	Could not determine dead time
7	one or more of P, I or D was out of range

Timeout Time (in sec): User programs Autotune timeout in seconds in this register. If EZ Series PLC can not finish autotuning within this time, the Autotune is aborted. User should program this field based on the dynamics of the process.

Tuning Type: User selects if PI or PID tuning is required.

PID Parameters

The PID Parameters consist of the Sample Rate, Setpoint, Proportional Gain, Integral (Reset) Time, Derivative (Rate) Time. The following section briefly describes each of these parameters .

Sample Rate - Enter the desired Sample Rate in this field. The Sample Rate is seconds and can be changed from 0.05 to 99.99 seconds.

PID Parameters	
Sample Rate (1/100 sec)	Base Reg + 0 (R1)
Setpoint	Base Reg + 1 (R2)
Proportional Gain	Base Reg + 2 (R3)
Integral (Reset) Time	Base Reg + 3 (R4)
Derivative (Rate) Time	Base Reg + 4 (R5)

***Note:** All numeric fields in this dialog box use Implied Decimal points. So to enter 0.05, you simply enter 5; the EZ Series PLC assumes two digits after the decimal point for most of the numeric entry fields, except where noted.

Setpoint - Enter the Setpoint in this field. This is the Setpoint used in the PID Loop calculation. The Setpoint is the desired process level.

Proportional Gain - Enter the Proportional Gain in this field. This is the gain of the proportional term of the PID equation. The valid range is 00.00 to 99.99. Setting this to zero removes the proportional term from the PID equation.

***Note:** The decimal point is implied. For example, "125" is 1.25. Default is 1.00

Integral (Reset) Time (T_i)- The units for this time are in seconds. The Valid range is 00.00 to 6000.0. This (along K_p and T_s) controls the integral term. Setting it to zero removes the integral term from the PID equation.

**Note: The decimal point is implied. For example, "125" is 12.5 seconds. Default is 0.3.*

Derivative (Rate) Time (T_d) - Enter the Derivative Gain in this field. This along with (T_s and K_p) makes the coefficient of the derivative term. The units are in seconds.

The valid range is 00.00 to 600.0. Setting this to zero removes the derivative term from the PID equation. **Note: The decimal point is implied. For example, "125" is 12.5 seconds. Default is 0.3*

PID Limits

The PID Limits consist of Deadband, Setpoint Low Limit, Setpoint High Limit, CV Low Limit, CV High Limit, CV Offset and Error. The following section briefly describes each of these.

Deadband - Enter the Deadband value in this field. This value is compared with the error value at loop update. If the absolute value of the error is less than the deadband value, then the error is considered as zero for PID computations.

PID Limits		
Deadband	0	Base Reg + 5 (R6)
Setpoint Low Limit	0	Base Reg + 6 (R7)
Setpoint HighLimit	0	Base Reg + 7 (R8)
CV Low Limit	0	Base Reg + 8 (R9)
CV HighLimit	4095	Base Reg + 9 (R10)
CV Offset	0	Base Reg + 10 (R11)
Error		Base Reg + 11 (R12)

Setpoint Low Limit - Enter the lower limit of your desired setpoint in this field. If the setpoint is below this value, then it will be set to the value you've entered in this field.

Setpoint High Limit - Enter the higher limit of your desired setpoint in this field. If the setpoint is above this value, then it will be set to the value you've entered in this field.

Control Value (CV) Low Limit - Enter the lower limit of the Control Value in this field. If the CV is below this value, then it will be set to the value you've entered in this field. Default is 0.

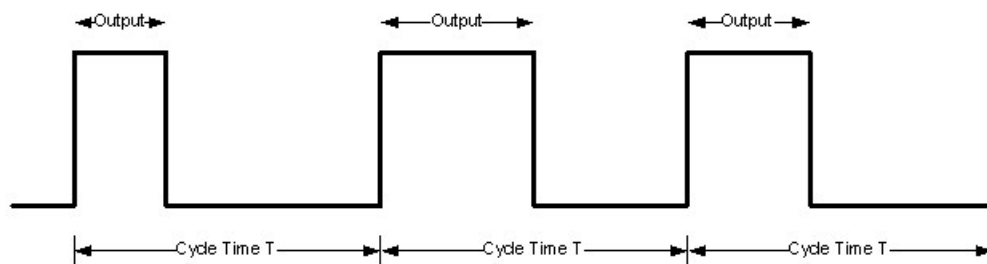
Control Value (CV) High Limit - Enter the higher limit of the Control Value in this field. If the CV is below this value, then it will be set to the value you've entered in this field. Default is 4095.

CV Offset - This is the constant offset that is added to the control variable. So, even when the Error is zero, the Control Variable equals offset.

Error – Shown on the dialog box for information only. EZ Series PLC used this register to store Error value.

Control Output

EZ Series PLC allows you to control a Digital output using PID control. The digital output provides a pulse out put on selected output address. The width of the pulse (within the cycle time) is proportional to the control value, as illustrated below:



Output is ON for time proportional to Control Value
 $\text{Output ON time} = (\text{CV}/4096) * T$

The following fields are programmed for the Digital Control Output: **Enable Control Output:** Check box to enable Digital Control Output. If the check box is unchecked, no digital output is provided.

☒ Enable Control Output	
Control Output Tag	01
Cycle Time (1/10 sec)	50
Min Duty Cycle (in %)	0
Max Duty Cycle (in %)	100

Base Reg + 14 (R15)
 (determined by CV Low)
 (determined by CV High)

Control Output Tag: Enter the discrete output address (O type) to provide Digital Control Output from the PID loop. The output module can be of any type (DC, AC or Relay type).

Cycle Time: Enter the Cycle time for the control output in tenths of a second. While selecting cycle time, keep in mind the load type that the output would be driving. For EM relays, we suggest that keep this time as high as possible to extend relay life.

Min Duty Cycle: This field is for display only. It is computed from the CV Low Limit $((\text{CV_LowLimit}/4096)*100)$ and expressed in percentage. As the name suggest, the output will remain on for minimum time even if the computed control value falls below the CV Low Limit.

Max Duty Cycle: This field is for display only. It is computed from the CV High Limit $((\text{CV_HighLimit}/4096)*100)$ and expressed in percentage. As the name suggest, the output will remain on for this maximum time even if the computed control value is above the CV high Limit.

Creating PID Tags

EZ Series PLC can automatically create tags corresponding to all the PID loop related variables (such as Sample Rate, SetPoint etc). To do so, perform the following steps:

1. Check the Create Tags checkbox (located beside the OK button) in the PID dialog.

☒ Create Tag	OK
--	----

2. Click the OK button. The following dialog box appears.

Create PID Tags

All the PID tags created will have the following naming convention:
 PID<loopNum>_<varName>
 where,
 <loopNum> = the loop number of the PID loop
 <varName> = the PID loop variable for which the tag is created. Use the table on the right to determine what the corresponding <varName> would be for a given PID variable.

For example, the tag for the Set Point variable for PID loop 2 will be created with the name PID2_SP. Also, note that certain tags, such as Cycle Time will only be created if they were specified in the PID dialog.

CAUTION
 If a tag with same name as a PID loop variable tag already exist, a new tag will NOT be created and the original tag will continue to exist in the tag database

Variable in Dialog	Corresponding <varName>
Sample Rate	Rate
Set Point	SP
Proportional Gain	KV
Integral (Reset) Time	KC
Derivative (Rate) Time	KP
Dead Band	DeadBand
Setpoint Low Limit	SPLow
Setpoint High Limit	SPHigh
CV Low Limit	CVLow
CV High Limit	CVHigh
CV Offset	CVOffset
Error	Error
PID Loop Mode	LoopMode
Anti-windup	AntiWindup
Use PV Square Root	UsePVSqrt
Saturation	Saturation
Start Autotune	StartAutoTune
Autotune Status	AutoTuneStatus
Timeout Time	TimeoutTime
Tuning Type	TuningType
Cycle Time	CycleTime

This dialog box tells you the naming convention that will be used for creating the PID loop tags. Note that all the tag names are fixed and denote the PID loop number the tag is associated with the variable the tag represents. Also, tags representing certain variables are only created if they were specified in the PID Loop (example, the tag representing Cycle Time).

- Click the Create Tags button. At this point all the tags are created and the results are displayed in a dialog as shown below. Note that if a tag already exists, that tag will not be created and it would be reported in this dialog. By clicking the Save button, you can save the this list of tags created and failed in a text file.

Created PID Tag Details

Successfully created tag(s):

- PID1_RATE
- PID1_SP
- PID1_KV
- PID1_KC
- PID1_KP
- PID1_DEADBAND
- PID1_SPLow
- PID1_SPHigh
- PID1_CVLow
- PID1_CVHigh
- PID1_CVOffset
- PID1_ERROR
- PID1_AUTOTUNESTATUS
- PID1_TIMEOUTTIME

PID Monitor

This section will explain how to setup and use the PID Monitor function within the EZPLC Editor. You can use this function to monitor and make real-time changes to your PID Loop. In order to use the PID Monitor function, you must be connected to the PLC. To begin, click on the EZPLC Menu and select PID Monitor. The dialog box below will appear. The various fields and parameters will be explained in the following pages.

PID Monitor

Select PID Loop: 1

Plot

	Current Value	New Value
☒ SetPoint		
☒ Control Variable (CV)		
☒ Process Variable (PV)		

Graph Parameters

Minimum SP	0
Maximum SP	0
Minimum CV	0
Maximum CV	4095
Interval(mSec)	1000
No of Ticks	25
☒ Show GridLines	

PID Parameters

	Current Value	New Value
Sample Rate		
Proportional Gain		
Integral		
Derivative		

CV can only be changed in the Manual mode

Mode: ☒ Auto ☐ Manual

Set Point (SP) & Process Variable (PV)

Control Variable (CV)

Min Sec mSec

Help Start Monitoring Apply New Values Freeze Graph Export to Excel Close

Select PID Loop - Use the drop arrow to select which PID Loop you would like to monitor (1 - 8).

Setpoint - This field displays the current value of your Setpoint. You can change the setpoint by entering a value in the New Value field and clicking the Apply New Values button at the bottom of the window.

Process Variable (PV) - This field displays the current value of the Process Variable (PV).

Control Variable (CV) - This field displays the current value of the Control Variable (CV).

Minimum SP - Enter the Minimum Setpoint value in this field.

Maximum SP - Enter the Maximum Setpoint value in this field.

***Note:** When selecting your values for Minimum and Maximum SP, it's a good idea to choose a number relatively close to the Process Variable. That way, when your graph is created you will be able to see more detail. The greater the range between your Minimum and Maximum SP, the less detail your graph will display. The shorter the range, the more detailed your graph will be. For this example,

the Process Value is at 550, so the Maximum SP is set at 575 and the Minimum SP is set for 525, leaving a range of 50 (25 above and below) to be displayed on the graph.

Minimum CV - Enter the Minimum Control Variable (CV) value in this field.

Maximum CV - Enter the Maximum Control Value (CV) value in this field.

Interval (mSec) - Enter the Interval value (in milliseconds) in this field.

No of Ticks - In this field, enter the Number of Ticks you would like to have displayed in the graph.

Show Grid Lines - Check this box if you would like Grid Lines to be displayed in your graph.

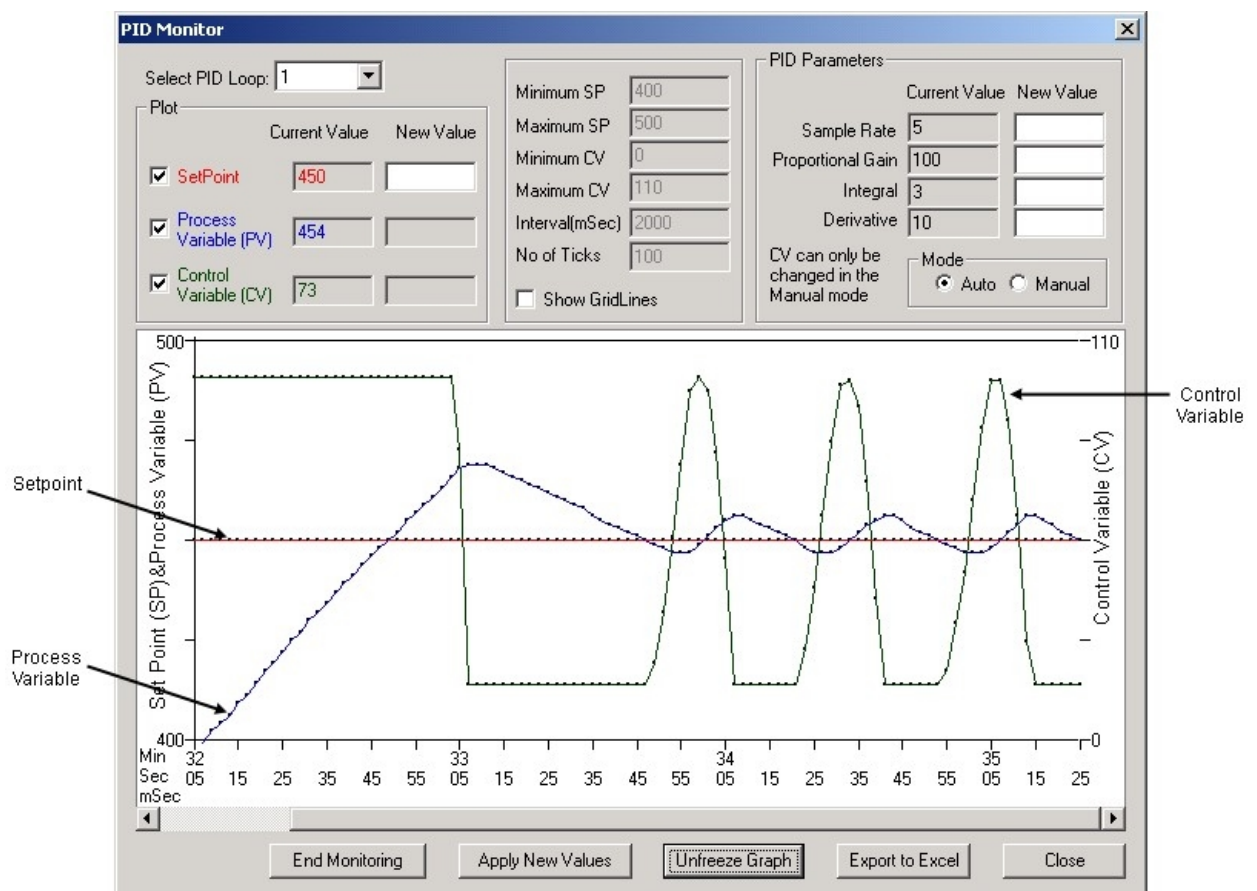
Sample Rate - In this field enter the Sample Rate to determine how often the PID Loop checks the process.

Proportional Gain - in this field enter the value of the Proportional Gain.

Integral - In this field enter the Integral value.

Derivative - In this field enter the Derivative value.

Mode - In this box you can choose Auto or Manual (you can only change the Control Variable in the Manual Mode). Once all of the parameters are defined, press the **Start Monitoring** button (on left) to begin monitoring your PID Loop. A graph will begin to appear as shown in the image below:



As you can see, the graph above has been created using the parameters explained on the previous page.

- The Setpoint and Process Variable were set to 450 and are represented in the graph by the line running through the middle of the graph.

- The Minimum SP of 400 is shown at the bottom left and the Maximum Limit of 500 is shown at the top left of the graph.
- The Control Variable was set to 110 and is represented on the right side of the graph.

The rest of the controlling buttons for PID Monitor are explained below:

End Monitoring - Press this button when you wish to stop the PID Monitor.

Apply New Values - Press this button once you have changed some of the parameters in PID Monitor and would like to begin monitoring those changes.

Freeze Graph - Press this button if you would like to see a still picture of the graph in its current state.

Export to Excel - Press this button to send all of the data within the graph to an Excel spreadsheet (you must have the Excel software installed onto your computer).

Close - Press this button stop the monitoring process and close the PID Monitor window.

PID Loop Auto Tuning

Proper selection of PID parameters, such as Proportion, Integral and Derivative coefficients, is important to get stable and responsive process control.

Many experienced users can estimate good starting values for the PID loops, and then tweak those to optimize PID loop performance, i.e. they can manually tune the process.

For those users who would like a help in estimating these starting values of P, I, and D coefficients, EZ Series PLC provides Autotune feature.

Please read following caution note carefully before attempting autotune.

CAUTION:

- **PID Loop control follows complex algorithm and its output depends on user-programmed PID parameters as well as the dynamics of the process being controlled.**
- **For safety reasons, users MUST ensure proper working of the PID before attempting autotune or before leaving loops in Auto mode. For example: Ensure that Process Value is correctly measured Ensure that the Control Variable moves the Process Variable in the right direction. You can set the Control Variable manually to check for this.**
- **During Autotune, users MUST observe the process closely and have access to emergency stops to stop the process if it goes out of control.**

Autotuning Pre-requisites

1. The EZ Series PLC must have firmware revision C.3 or later.
2. PID loops must be configured.

Autotune Control

Each PID Loop is controlled by the Start Autotune discrete variable (which is at Discrete Base+4). If the variable goes from false to true, and the loop is in manual mode, EZ Series PLC would start autotuning that loop.

Autotune Setup

The EZ Series PLC can autotune PID loops, i.e. it can estimate the values for the Proportional Gain, Integral (Reset) time, and Derivative (Rate) time for PID loop. The dialog box allows you to setup the loop for autotune. EZ Series PLC uses Ziegler-Nichols method to estimate the PID parameters. The following are the setup parameters for Autotune:

Start Autotune - Shown on the dialog box for information only. The Start Autotune discrete is at Discrete Base+4. EZ Series PLC initiates autotuning of a loop when this bit transitions from 0 to 1. Autotuning of the loop is started regardless of the selected "PID Loop Mode" of the loop. Once Autotune is started, you can stop it by making setting this bit to 0.

Start Autotune	Discrete Base + 4 (S5)
Autotune Status	Base Reg + 12 (R13)
Timeout Time (in sec) 600	Base Reg + 13 (R14)
Tuning Type ☒ PID ☐ PI	Discrete Base + 5 (S6) 0 = PID, 1 = PI

Note: Autotune is performed by EZ Series PLC observing open loop response to a step change in the control value. Before starting autotune, the process should be in a steady state. During Autotune, watch the process variable closely for it to be within the safe limits.

Autotune Status: Shown on the dialog box for information only.

During Autotune, EZ Series PLC reports the status of Autotune in the register.

Register Value	Description
0	Tuning in progress
1	Tuning Done
2	User canceled tuning
3	Control Value could not be incremented
4	The tuning algorithm failed to fit the curve
5	Division by zero error
6	Could not determine dead time
7	one or more of P, I or D was out of range

Timeout Time (in sec): User programs Autotune timeout in seconds in this register. If EZ Series PLC can not finish autotuning within this time, the Autotune is aborted. User should program this field based on the dynamics of the process.

Tuning Type: User selects if PI or PID tuning is required.

Autotuning Loops

To Autotune one or more loops, please follow below given steps:

1. Make sure you have read the caution at the beginning of this topic.
2. Set the PID Loop Mode to Manual for all the loops that require autotune.
3. To start Autotuning a loop, set the Start Autotune discrete variable to ON.
4. EZ Series PLC would start Autotune by changing Control Variable (CV), measuring the impact on the Process Variable (PV) of this change.
5. EZ Series PLC estimates the P, I, D gains using the Ziegler-Nichols open loop two point tuning method.
6. EZ Series PLC Editor can be used to monitor CV and PV while EZ Series PLC is autotuning loops.

