EZAutomation

6

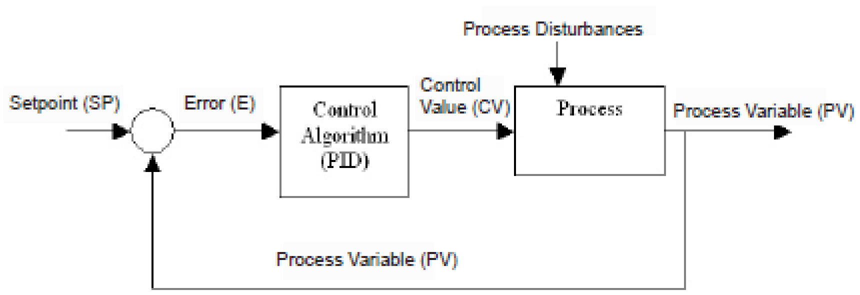
Chapter 6: PID Loop

In this Chapter...

6.1 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 EZLogix

EZLogix 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 EZLogix Designer Pro software in online mode.

PID Algorithms used in EZLogix

The EZLogix 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

The Error is computed as follows:

$E_n = PV_n - SP_n$ for Direct Acting
$= SP_n - PV_n$ for Reverse Acting

Then the CV_n is computed as follows:

Position Algorithm:

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

Velocity Algorithm:

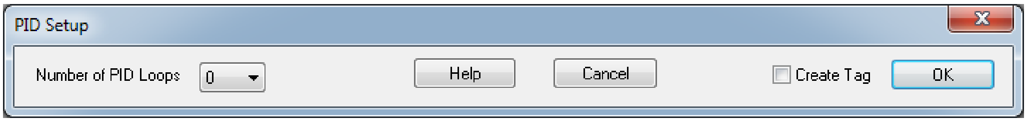
$$CV_n = K_p \times \left[E_n + (T_s/T_i) \times \sum_{i=0}^{n} E_i + (T_d/T_s) \times (PV_n - PV_{n-1}) \right] + CV_0$$

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.

6.2 PID Setup

The following section will explain how to setup a PID loop using your EZLogix Designer Pro software. To access the PID Setup, perform the following steps:

1. 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).

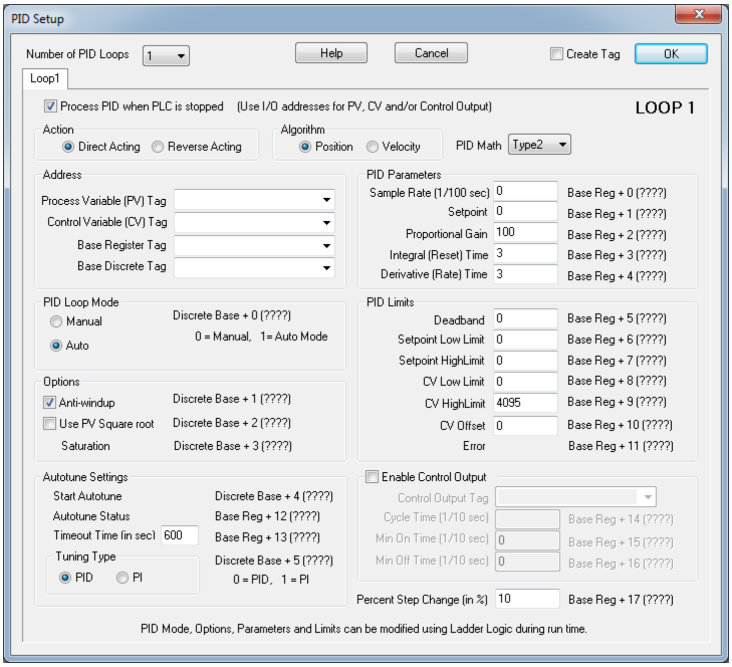


PID Setup

Number of PID Loops: 0

Help | Cancel | ☐ Create Tag | OK

2. 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).
3. As soon as you select a number of loops other than 0, the following dialog box will appear:



PID Setup

Number of PID Loops: 1

Help | Cancel | ☐ Create Tag | OK

Loop1

☑ Process PID when PLC is stopped (Use I/O addresses for PV, CV and/or Control Output)

LOOP 1

Action: [✓ Direct Acting] Reverse Acting

Algorithm: [✓ Position] Velocity

PID Math: Type2

Address

Process Variable (PV) Tag	
Control Variable (CV) Tag	
Base Register Tag	
Base Discrete Tag	

PID Parameters

Sample Rate (1/100 sec)	0	Base Reg + 0 (????)
Setpoint	0	Base Reg + 1 (????)
Proportional Gain	100	Base Reg + 2 (????)
Integral (Reset) Time	3	Base Reg + 3 (????)
Derivative (Rate) Time	3	Base Reg + 4 (????)

PID Loop Mode

Manual [✓ Auto]

Discrete Base + 0 (????)

0 = Manual, 1= Auto Mode

PID Limits

Deadband	0	Base Reg + 5 (????)
Setpoint Low Limit	0	Base Reg + 6 (????)
Setpoint HighLimit	0	Base Reg + 7 (????)
CV Low Limit	0	Base Reg + 8 (????)
CV HighLimit	4095	Base Reg + 9 (????)
CV Offset	0	Base Reg + 10 (????)
Error		Base Reg + 11 (????)

Options

☑ Anti-windup	Discrete Base + 1 (????)
☐ Use PV Square root	Discrete Base + 2 (????)
Saturation	Discrete Base + 3 (????)

Autotune Settings

Start Autotune		Discrete Base + 4 (????)
Autotune Status		Base Reg + 12 (????)
Timeout Time (in sec)	600	Base Reg + 13 (????)

Tuning Type: [✓ PID] PI

Discrete Base + 5 (????)

0 = PID, 1 = PI

☐ Enable Control Output

Control Output Tag		
Cycle Time (1/10 sec)		Base Reg + 14 (????)
Min On Time (1/10 sec)	0	Base Reg + 15 (????)
Min Off Time (1/10 sec)	0	Base Reg + 16 (????)

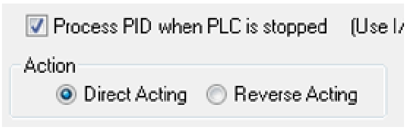
Percent Step Change (in %) 10 Base Reg + 17 (????)

PID Mode, Options, Parameters and Limits can be modified using Ladder Logic during run time.

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 EZLogix PLC then maps the next 31 registers automatically, making a total of 32 registers per block. Out of the block of 32 registers, 17 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 EZLogix 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 (Use I/
Action
● Direct Acting ○ Reverse Acting

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.

**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.*

Controller Action – To determine whether the Controller Action needs set as Direct Acting or Reverse Acting, it is helpful to understand the difference between "Process Action" and "Controller Action." First, ascertain what type of process (direct acting or reverse acting) best describes your system.

- Direct Acting Process: 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) or a decrease in the heater's output will result in a decrease in temperature.
- Reverse Acting Process: The CV and PV move in opposite direction such as an increase in CV decreases the PV or vice versa.
  - For example: In an air conditioning or cooling application, more power is applied to create a reduction or decrease in the temperature.

For simple systems, after identifying the type of process that describes your system, set the Controller Action as the opposite of the process. For instance, if your system utilizes a direct acting process, the Controller Action will need set as Reverse Acting. If your system utilizes a reverse acting process, the Controller Action will need set as Direct Acting. (See below for further explanation.)

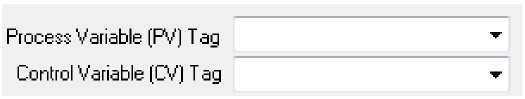
- Controller Action = Reverse Acting: The controller monitors PV. As PV decreases, the Controller will increase CV and vice versa.
  - For example: In the heating application listed above (a direct acting process), the controller would need set as Reverse Acting in order to manage the

temperature in relation to the setpoint. So that if the temperature decreases due to outside variables, the controller will increase the heater to restore the temperature back to desired level. Conversely, if the temperature increases due to outside variables, the controller will decrease the heating output to restore the temperature back to the established setpoint.

- Controller Action = Direct Acting: The controller monitors PV. As PV increases the controller will increase CV and vice versa.
  - In the example of the cooling application (or the reverse acting process), the controller would need set as Direct Acting. Therefore, if the temperature increases due to outside variables, the controller will increase the cooling power in order to restore the temperature to desired level. Conversely, if the temperature decreases due to outside variables, the controller will decrease the cooling output to restore the temperature back to the established setpoint.

EZLogix computes error term, based on this choice, as follows:
$E_n = PV_n - SP_n$ for Direct Acting
$E_n = SP_n - PV_n$ for Reverse Acting

Algorithm (Position or Velocity) – EZLogix supports two PID algorithms, known as Position and Velocity algorithms. Select whether you would like to use a Position math equation or a Velocity math equation.

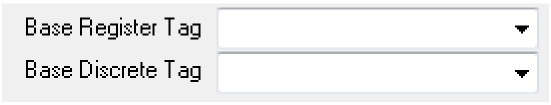


Process Variable (PV) Tag - Use the drop-down arrow or enter a tag address where you would like the Process Variable to be stored. You can use R or IR register types. If you use an IR type tag, then you are reading the Process Variable directly from an Input Module. If use an R-type after some scaling)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

**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.*

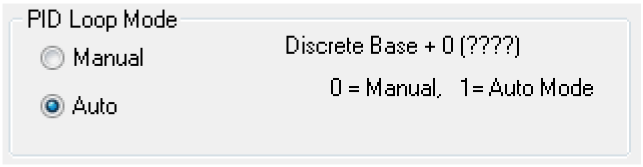
Control Variable (CV) Tag - Use the drop-down arrow or enter a tag address where you would like the Control Variable to be stored. The CV tag has the flexibility of using R or OR registers, If you use OR, then EZLogix writes the CV directly to an Output Module. If you use the R type for CV tag, you will 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 Discrete Tag

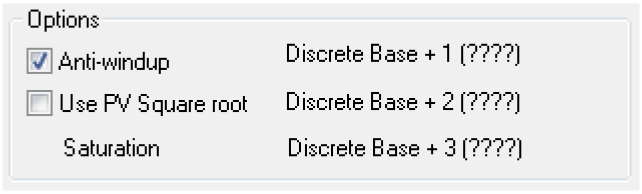
Base Register Tag - Base Register Tag/Address defines the starting address of a Contiguous Block of 32 registers that are used to store PID Parameters and Status information. Please see the dialog box to find the addresses of desired parameter within the block.

Base Discrete Tag - Base Discrete Tag/Address defines the starting address of a Contiguous Block of 8 discretes that are used to store PID Parameters and Status information.


PID Loop Mode
Manual [✓ Auto]
Discrete Base + 0 (????)
0 = Manual, 1= Auto Mode

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 (EZLogix>PID Monitor) can be used to modify Control Variable in manual 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.

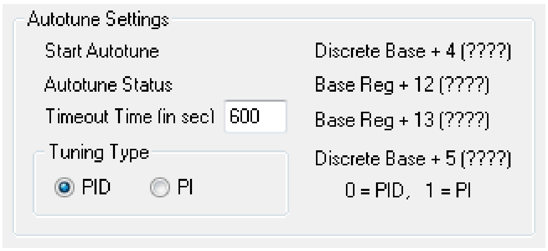

Options

✓ Anti-windup	Discrete Base + 1 (????)
☐ Use PV Square root	Discrete Base + 2 (????)
Saturation	Discrete Base + 3 (????)

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.

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 Settings

Start Autotune — Discrete Base + 4 (????)

Autotune Status — Base Reg + 12 (????)

Timeout Time (in sec) 600 — Base Reg + 13 (????)

Tuning Type — Discrete Base + 5 (????)

(● PID) ○ PI — 0 = PID, 1 = PI

Autotune Setup

The EZLogix 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. EZLogix uses Ziegler-Nichols method to estimate the PID parameters. Following are the setup parameters for Autotune:

Note: Autotune is performed by EZLogix observing open loop response to a 10% step change in the control value. Before starting autotune, the process should be in a steady state. For best results, the steady state should be within 10% of the operating set point. During Autotune, watch the process variable closely for it to be within the safe limits.

Start Autotune - Shown on the dialog box for information only.
The Start Autotune discrete is at Discrete Base+4. EZLogix 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 cancel it by setting this bit to 0. When Autotune is finished, either from success, error, or user canceled, setting this bit from 0 to 1 will return the Autotune to an idle state freeing any memory allocated for the Autotune.

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

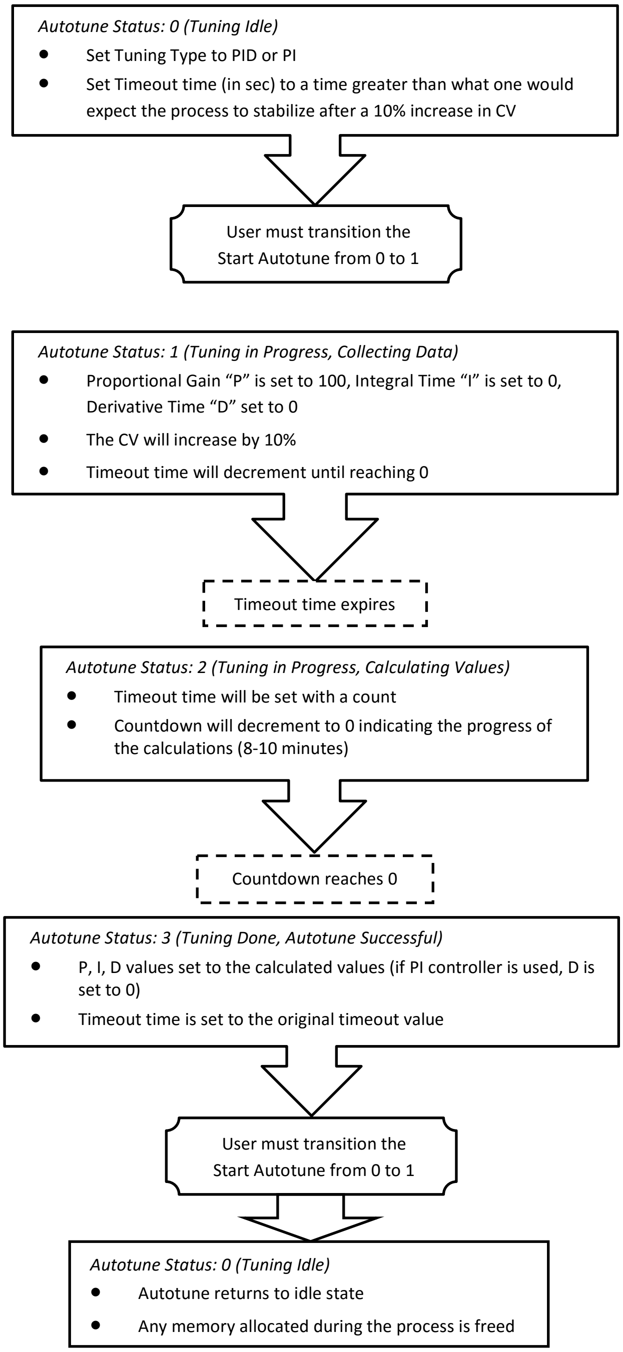
Register Value	Description
0	Tuning Idle
1	Tuning in progress, collecting data
2	Tuning in progress, calculating values
3	Tuning done, Autotune successful
4	User canceled tuning
5	Control Value could not be incremented
6	The tuning algorithm failed to fit the curve
7	Division by zero error
8	Could not determine dead time
9	One or more of P, I, or D was out of range
10	Tuning low memory error

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

- When Autotune Status = 1 (collecting data), this will time down to 0 to indicate the progress of the data collection.
- While Autotune Status = 2 (calculating values), this register will be used as a decrementing counter counting down to 0 indicating the progress of the calculations

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

Normal Autotune Sequence



Autotune Status: 0 (Tuning Idle)

- Set Tuning Type to PID or PI
- Set Timeout time (in sec) to a time greater than what one would expect the process to stabilize after a 10% increase in CV

User must transition the Start Autotune from 0 to 1

Autotune Status: 1 (Tuning in Progress, Collecting Data)

- Proportional Gain “P” is set to 100, Integral Time “I” is set to 0, Derivative Time “D” set to 0
- The CV will increase by 10%
- Timeout time will decrement until reaching 0

Timeout time expires

Autotune Status: 2 (Tuning in Progress, Calculating Values)

- Timeout time will be set with a count
- Countdown will decrement to 0 indicating the progress of the calculations (8-10 minutes)

Countdown reaches 0

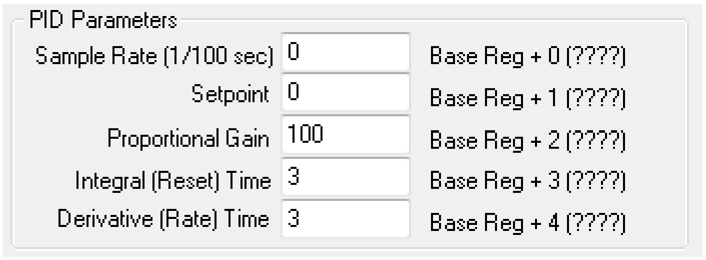
Autotune Status: 3 (Tuning Done, Autotune Successful)

- P, I, D values set to the calculated values (if PI controller is used, D is set to 0)
- Timeout time is set to the original timeout value

User must transition the Start Autotune from 0 to 1

Autotune Status: 0 (Tuning Idle)

- Autotune returns to idle state
- Any memory allocated during the process is freed


PID Parameters

Sample Rate (1/100 sec)	0	Base Reg + 0 (????)
Setpoint	0	Base Reg + 1 (????)
Proportional Gain	100	Base Reg + 2 (????)
Integral (Reset) Time	3	Base Reg + 3 (????)
Derivative (Rate) Time	3	Base Reg + 4 (????)

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.

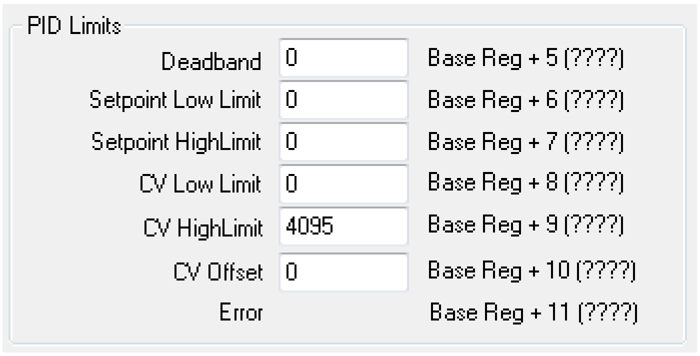
Sample Rate - Enter the desired Sample Rate in this field. The Sample Rate is seconds and can be changed from 0.05 to 99.99 seconds.
**Note: All numeric fields in this dialog box use Implied Decimal points. So to enter 0.05, you simply enter 5; the EZLogix assumes two digits after the decimal point for 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

Deadband	0	Base Reg + 5 (????)
Setpoint Low Limit	0	Base Reg + 6 (????)
Setpoint HighLimit	0	Base Reg + 7 (????)
CV Low Limit	0	Base Reg + 8 (????)
CV HighLimit	4095	Base Reg + 9 (????)
CV Offset	0	Base Reg + 10 (????)
Error		Base Reg + 11 (????)

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.

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.\

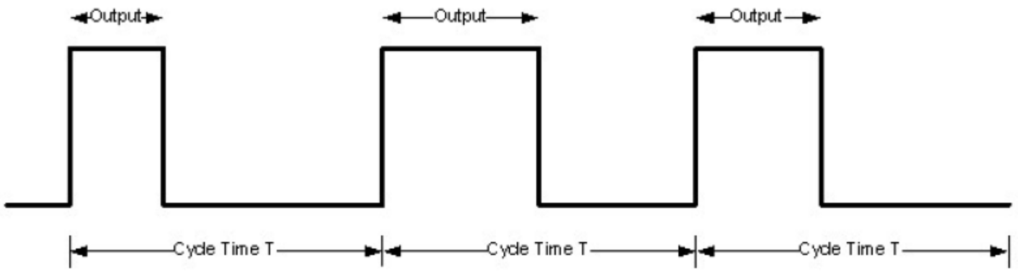
Control Value (CV) High Limit - Enter the higher limit of the Control Value in this field. If the CV is below this value, then it will be set to the value you've entered in this field. Default is 4095.

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. EZLogix PLC used this register to store Error value.

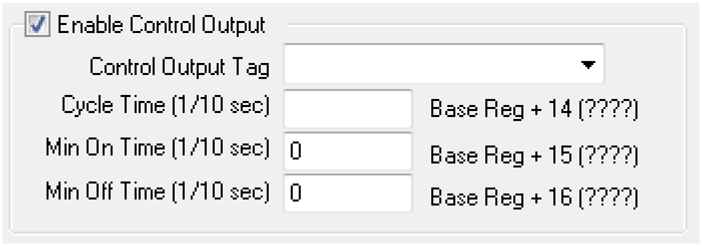
Control Output

EZLogix 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
Output ON time = (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		
Cycle Time (1/10 sec)		Base Reg + 14 (????)
Min On Time (1/10 sec)	0	Base Reg + 15 (????)
Min Off Time (1/10 sec)	0	Base Reg + 16 (????)

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).

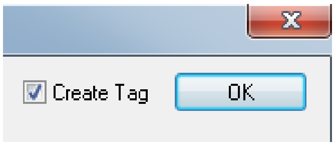
Cycle Time: Enter the Cycle time for the control output in tenths of a second. While selecting cycle time, keep in mind the load type that the output would be driving. For EM relays, we suggest that keep this time as high as possible to extend relay life.

Min Duty Cycle: This field is for display only. It is computed from the CV Low Limit ((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 ((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

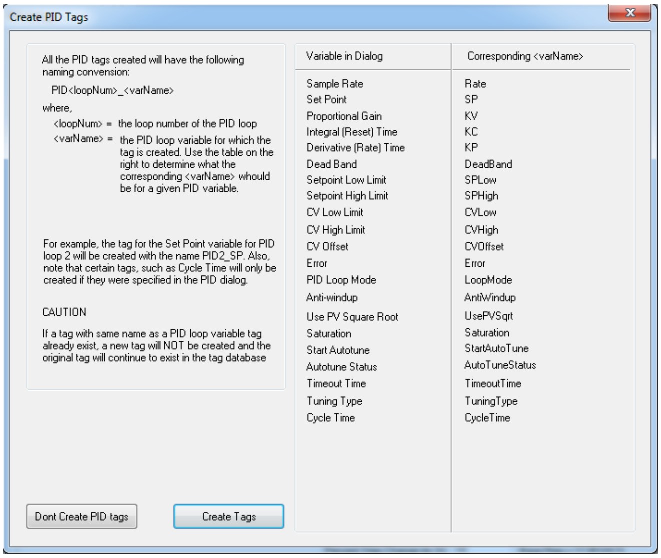
Creating PID Tags

EZLogix 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.

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> whould 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.

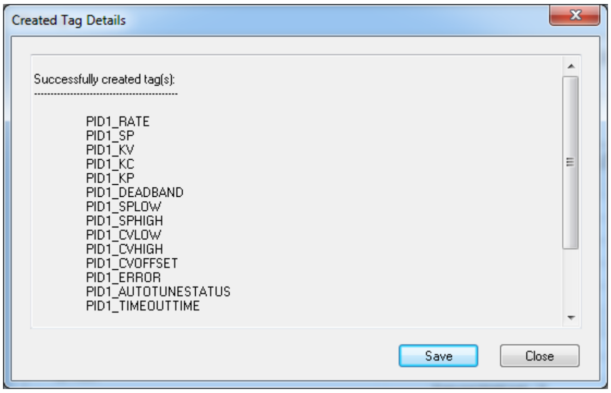
CAUTION

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).

3. 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 this list of tags created and failed in a text file.



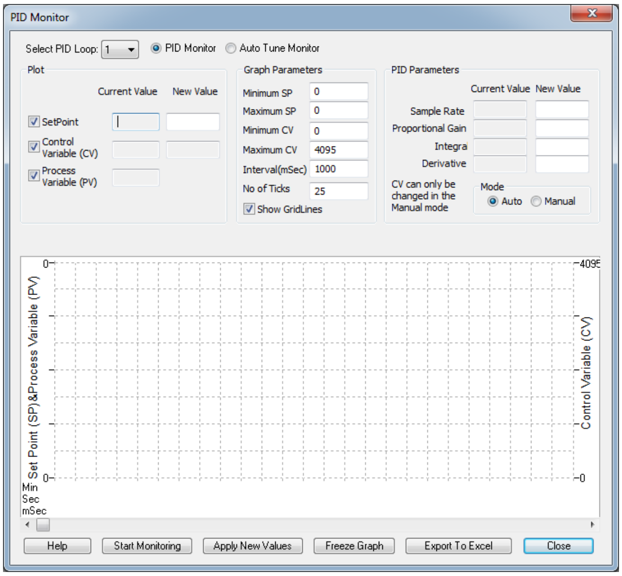
Created Tag Details

```
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
```

6.3 PID Monitor

This section will explain how to setup and use the PID Monitor function within the EZLogix Designer Pro. 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 to the PLC Menu and select PID Monitor (as shown to the left). The dialog box below will appear. The various fields and parameters will be explained in the following pages.


PID Monitor

Select PID Loop: 1 [✓ PID Monitor] Auto Tune Monitor

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		
Integra		
Derivative		

CV can only be changed in the Manual mode

Mode [✓ Auto] Manual

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.

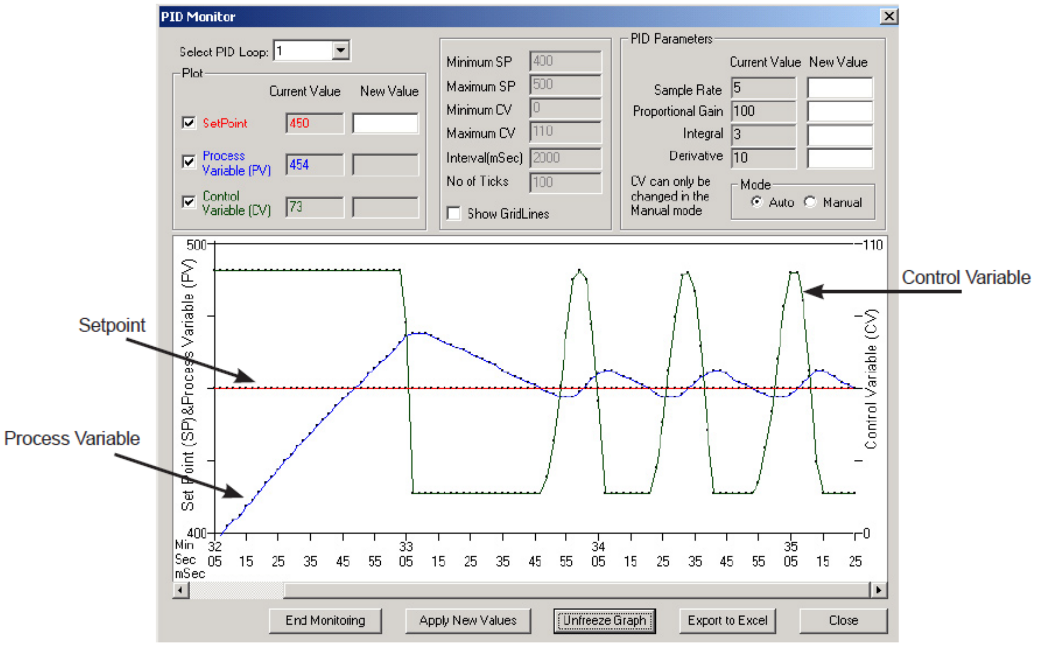
Integral - In this field enter the Integral value.

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).

Start Monitoring

Once all of the parameters are defined, press the Start Monitoring button (shown to the left) to begin monitoring your PID Loop. A graph will begin to appear as shown in the image below:


PID Monitor

Select PID Loop: 1

Plot

	Current Value	New Value
☑ SetPoint	450	
☑ Process Variable (PV)	454	
☑ Control Variable (CV)	73	

Minimum SP	400
Maximum SP	500
Minimum CV	0
Maximum CV	110
Interval(mSec)	2000
No of Ticks	100

☐ Show GridLines

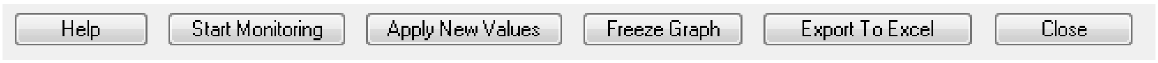
PID Parameters

	Current Value	New Value
Sample Rate	5	
Proportional Gain	100	
Integral	3	
Derivative	10	

CV can only be changed in the Manual mode

Mode: [✓ Auto] Manual

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 on the next page.



End Monitoring / Start Monitoring - Press this button when you wish to stop / start 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.