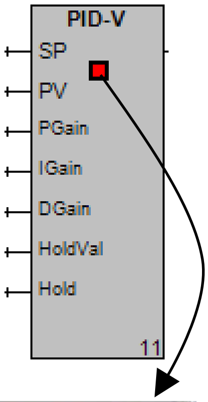
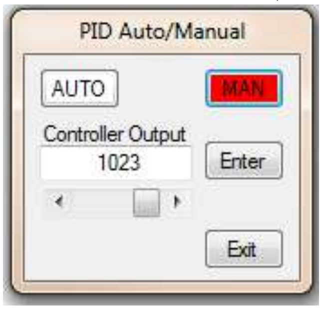
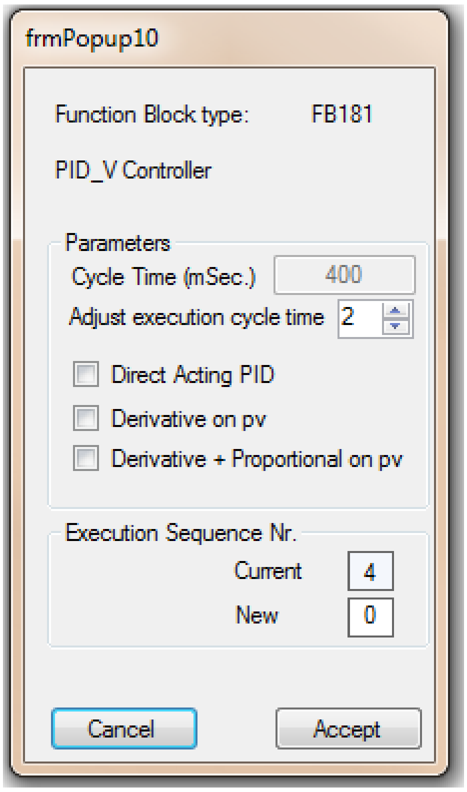

Function Description

This function block provides proportional, integral and derivative action on an error signal derived from the process variable (PV) and setpoint (SP) inputs. Inputs are provided for proportional (PGain), integral (IGain), and derivative (DGain) factors. A 'hold' facility allows the controller output to track the signal at the input HoldVal. Manual control of the controller is provided when on-line. A click on the red rectangle on the function block will pop up the Auto/Manual GUI, which provides switching between auto and manual modes and adjusting of the output when in manual mode. For the hold and manual modes, return to auto control is bumpless. The hold mode takes priority over the manual mode.







Popup Parameters

Cycle Time (mSec):

Indicates the cycle time assigned to this PID function block. It can be adjusted in
multiples (max 50) of the execution time of the task hosting this block

Direct acting PID:

(selected) Error = PV - SP (increase in PV will increase controller output)
(unselected) Error = SP – PV (increase in PV will decrease controller output)

Derivative on pv:

(selected) Derivative term calculated using process variable (PV)

Derivative and Proportional on pv:

(selected) Both derivative and proportional terms calculated using process variable (PV).

Execution Sequence Nr.

Allow modification of execution sequence.

Input/Output and Parameters

Type	Description	Data Type	Range
SP	Set Point	INT	0 ----1023 See Note 1
PV	Process Variable	INT	0 ----1023 See Note 1
PGain	Proportional constant	FP32	FP32 range
IGain	Integral factor	FP32	FP32 range
DGain	Derivative factor	FP32	FP32 range
HoldVal	Output override value	INT	0 ----1023
Hold	Output override switch	Boolean	0 or 1
PID Output	Controller output	INT	0 ----1023

Note 1: Exceeding this value can cause unpredicted results.

It is required that Function Block 181 is executed at regular intervals; therefor the editor will only allow this function block to be placed in code pages assigned to Time Tasks.

Algorithm Description

The continuous form of the standard PID equation is given by:

$$M = Kc(e + \frac{1}{T_I}\int edt + T_D \frac{de}{dt}) \quad ....1$$

The discrete equivalent of the above equation that will compute the controller output value at time i is given by:

$$m_i = Kc[e_i + \frac{\Delta T}{T_I}\sum_{0}^{i} e_i + \frac{T_D}{\Delta T}(e_i - e_{i-1})] \quad ....2$$

By subtracting two successive values (i.e. $\Delta m_i = m_i - m_{i-1}$) we obtain an equation that calculates the incremental change in controller output over a time period ΔT :

$$\Delta m_i = Kc[e_i - e_{i-1} + \frac{\Delta T}{T_I} e_i + \frac{T_D}{\Delta T}(e_i - 2e_{i-1} + e_{1-2})] \quad ....3$$

The value Δm_i is then used as the input to an integrator to arrive at the actual output value for the function block.

It must be noted that the selection of pv used for derivative, or used for derivative and proportional calculation, together with selection of direct/reverse action, require different versions of the equation in 3 above. These changes are taken care of by the compiler in VPS_P18 based on the user's selection choices on the configuration popup. The merits of the different arrangements will not be discussed here, although for information the details of the different equations are given.

Equation 3 is used for direct (e = PV – SP) and reverse (e = SP – PV) acting controllers when calculating the proportional, integral and derivative terms using the error (e) value.

Equation 4 is used for direct (e = PV – SP) acting controllers when calculating the proportional and integral terms using the error (e) value, and the derivative term using the process measurement (p) value.

$$\Delta m_i = Kc[e_i - e_{i-1} + \frac{\Delta T}{T_I} e_i + \frac{T_D}{\Delta T}(p_i - 2p_{i-1} + p_{i-2})] \quad ....4$$

Equation 5 is used for reverse (e = SP – PV) acting controllers when calculating the proportional and integral terms using the error (e) value, and the derivative term using the process measurement (p) value.

$$\Delta m_i = Kc[e_i - e_{i-1} + \frac{\Delta T}{T_I} e_i - \frac{T_D}{\Delta T}(p_i - 2p_{i-1} + p_{i-2})] \quad ....5$$

Equation 6 is used for direct (e = PV – SP) acting controllers when calculating the integral term using the error (e) value, and the proportional and derivative terms using the process measurement (p) value.

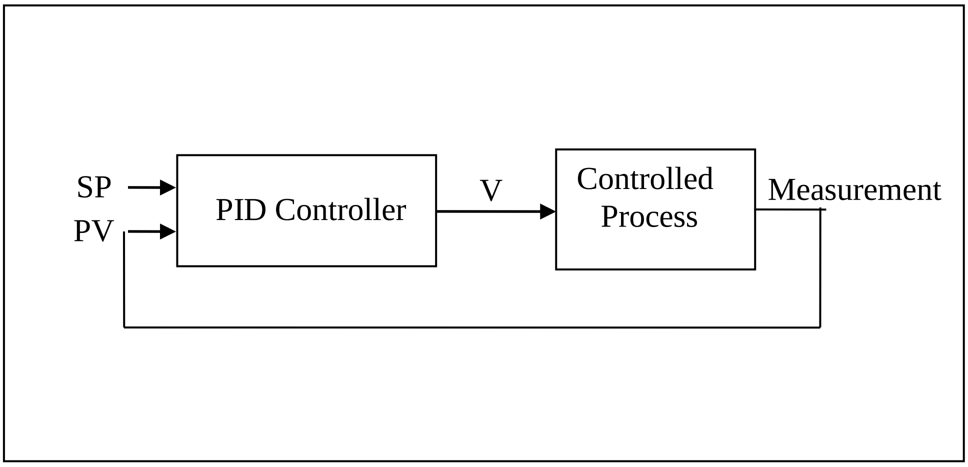
$$\Delta m_i = Kc[p_i - p_{i-1} + \frac{\Delta T}{T_I} e_i + \frac{T_D}{\Delta T}(p_i - 2p_{i-1} + p_{i-2})] \quad ....6$$

Equation 7 is used for reverse (e = SP – PV) acting controllers when calculating the integral term using the error (e) value, and the proportional and derivative terms using the process measurement (p) value.

$$\Delta m_i = Kc[p_{i-1} - p_i + \frac{\Delta T}{T_I} e_i - \frac{T_D}{\Delta T}(p_i - 2p_{i-1} + p_{i-2})] \ldots\ldots\ldots\ldots\ldots\ldots 7$$

In the equations above values Kc, T_I and T_D characterize the process being controlled, and the more accurate we know them the better we can set up (tune) the PID to control the process. A method for determining their values and how to use them to obtain the input values for PGain, IGain and DGain is discussed below.

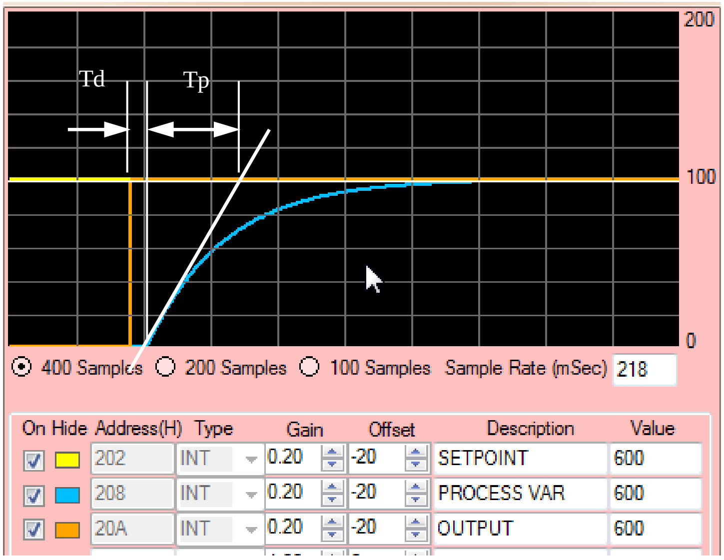
PID's are used to control temperature, speed, liquid level and flow, to name but a few. The basic arrangement of these control loops are given in the following figure, where the objective is to automatically adjust the PID output V so as to maintain PV equal to SP even when the controlled process change. For example to maintain a desired speed even when the load on the controlled motor change.



Notes on determining the values of PGain, IGain and DGain.

Ziegler and Nichols conducted experiments and proposed equations for determining the values of Kc, T_I and T_D by using a set of characteristics based on the transient step response of the controlled device or process. These characteristics are Kp (process gain), Tp (process time constant) and Td (dead time). We will use here their equations that apply to devices or processes that will stabilize after a step change. That means the unit-step response will resemble an S-shaped curve with no overshoot. This is also referred to as the reaction-curve method.

One way of obtaining the reaction curve is to use VPS_P18's on-line trend view tool to display the PID output and the process feedback. The PID is put into manual mode by making use of the Auto/Manual GUI. The following is a screen shot of the results obtained when changing the PID output from 0 to 600, and shows how to determine the dead-time Td, and process time-constant Tp. Best results are obtained when the step response is carried out near the normal operating point of the control loop.



The process gain Kp is calculated as:

Kp = (% change in measured value)/ (% change in PID output)

These values are now used to calculate the values Kc, T_I and T_D using the applicable Ziegler-Nichols tuning equations:

Type of controller			
P	$Kc = \dfrac{Tp}{Kp * Td}$		
PI	$Kc = \dfrac{0.9Tp}{Kp * Td}$	$T_I = 3.33Td$	
PID	$Kc = \dfrac{1.1Tp}{Kp * Td}$	$T_I = 2.0Td$	$T_D = 0.5Td$

The values required at the PGain, IGain and DGain inputs of FB181 can now be calculated as:

PGain = Kc ……………………………………8

IGain = $\dfrac{\Delta T}{T_I}$ ……………………………….9

DGain = $\dfrac{T_D}{\Delta T}$ ……………………………….10

Where ΔT is the cycle time (in Seconds) of the PID function block. Note that the cycle time of this block can be adjusted in multiples of the execution time of the task in which it is hosted.

The above calculated parameters will typically give you a response with about 25% overshoot and good settling time. In many cases we require less overshoot at the cost of rise time. The following table can be very handy when adjusting the parameters; it indicates the effect of increasing each of the 3 parameters.

Parameter	Rise Time	Overshoot	Settling Time
PGain	Decrease	Increase	N/A
IGain	Decrease	Increase	Increase
DGain	N/A	Decrease	Decrease

Application example

The following show a screen shot of a test application in on-line mode. The user can implement this to investigate the behavior of the PID function block FB181. The 'controlled process' is simulated by a Dead-Time-Delay (FB177) and a 1st-Order-Lag (FB176). The values for PGain, IGain, and DGain are obtained from EE-Read (FB45) function blocks, while a Read-Connector supplies the setpoint value. All these values can be adjusted on-line as indicated by the red rectangle in these blocks. Using EE-Read function blocks for the tuning parameters means you do not lose your settings when power is removed from the PIC processor. This application can also run in the VPS_P18 real-time simulator.