

ПИД регулятор в Sysmac Studio. Описание.

1.Описание математической модели регулятора

Математическая модель ПИД регулятора, используемого в стандартном ФБ PIDAT в среде Sysmac Studio:

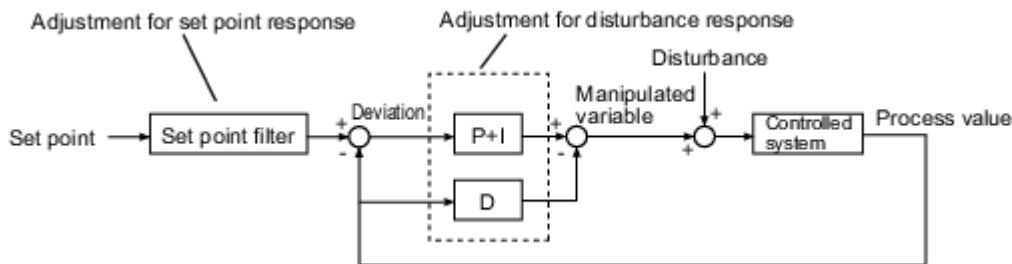


Рис.1 – Обобщенная структурная схема ПИД-регулятора

На схеме:

Set point – задание для регулятора;

Deviation – отклонение (ошибка, рассогласование), это разность между сигналом заданием и сигналом обратной связи (получаемого от датчика);

Manipulated variable – выход регулятора, управляющий объектом сигнала;

Disturbance – возмущающее воздействие;

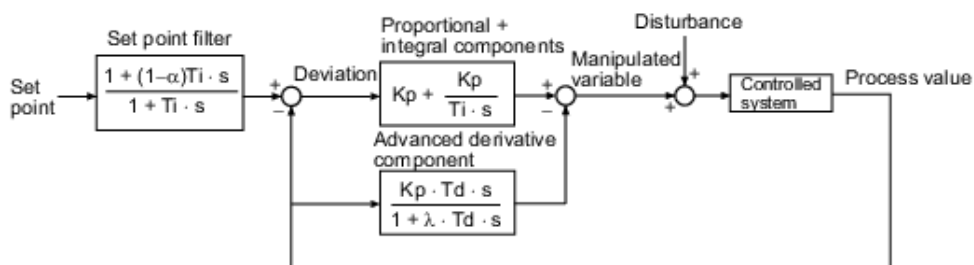
Controlled system – объект управления;

Process value – измеряемое значение регулируемого параметра (выход объекта), он же сигнал обратной связи;

P, I, D – пропорциональная, интегральная, дифференциальная составляющие регулятора;

Set point filter – фильтрующее звено для сигнала задания.

Ниже показана та же структурная схема, но в развернутом виде:



K_p: Proportional constant
T_i: Integration time
T_d: Derivative time
s: Laplace operator
α: 2-PID parameter
λ: Incomplete derivative coefficient

Рис.2 – Детальная структурная схема ПИД-регулятора

На схеме:

K_p – коэффициент усиления пропорциональной части;

T_i – постоянная времени интегрирования;

T_d – постоянная времени дифференцирования;

s – оператор Лапласа (дифференцирования);

α – коэффициент фильтра сигнала задания 0..1 (позволяет делать регулятор одинаково хорошо обрабатывающим скачки как сигнала задания, так и сигнала возмущения);

λ – коэффициент неполной производной в усовершенствованном дифференцирующем звене.

Из приведенных схем видно, что ПИД-регулятор отличается от классического вида тем, что содержит блок фильтра для сигнала задания, а также тем, что имеет нестандартную дифференциальную часть.

2.Описание интерфейса функционального блока PIDAT

Рассмотрим применение встроенного ФБ PIDAT. Общий вид функционального блока типа PIDAT:

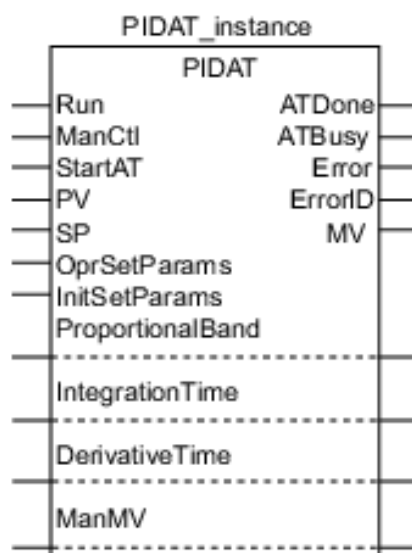


Рис. 3 – Общий вид функционального блока PIDAT

Входы блока:

Название	Расшифровка	Тип	Значение/ описание	Тип данных
Run	Флаг запуска	Input	TRUE: Выполнять FALSE: Остановить	BOOL
ManCtl	Ручное/автомат управление выходом		TRUE: Ручное FALSE: Автомат	BOOL

StartAT	Старт автонастройки параметров ПИД		TRUE: Выполнить FALSE: Отменить	BOOL
PV	Обратная связь от датчика объекта		-	REAL
SP	Уставка (задание) регулятора		-	REAL
OprSetParams	Структура – параметры работы		Параметры настраиваются в процессе работы	_sOPR_SET_PARAMS
InitSetParams	Структура – начальные настройки		Начальные настройки	_sINIT_SET_PARAMS

Входы-выходы блока:

Название	Расшифровка	Тип	Значение	Тип данных
Proportional band	Пропорциональный коэффициент	In-out	-	REAL: 0.01 to 1000.00
Integration time	Постоянная времени интегрирования		Увеличение значения ослабляет интегральную часть регулятора. При нуле интегратор отключен.	TIME: T#0.0000 s to T#10000.0000 s
Derivative time	Постоянная времени дифференцирования		Увеличение значения усиливает дифференциальную часть. При нуле диф. часть отключена.	TIME: T#0.0000 s to T#10000.0000 s
Manual manipulated variable	Ручной выход		Управляемый вручную выход	REAL: -320 to 320

Выходы:

Название	Расшифровка	Тип	Значение	Тип данных
ATDone	Флаг нормального завершения автонастройки	Output	TRUE: нормальное завершение FALSE: ошибка автонастройки	BOOL
ATBusy	Автонастройка в процессе		TRUE: Автонастройка FALSE: Автонастройки нет	
MV	Ручной выход		Управляемый вручную выход	REAL: -320 to 320

Описание структур с параметрами **OprSetParams, InitSetParams**

Структура **_sOPR_SET_PARAMS**

Название	Описание	Тип данных	Диапазон	Ед.из м.	По умолч.
OprSetParams	Параметры которые устанавливаются в процессе работы	_sOPR_SET_PARAMS	-		
MVLowLmt	Нижний предел ручного выхода	REAL	-320 ... 320	%	0
MVUpLmt	Верхний предел ручного выхода	REAL			100
ManResetVal	Значение ручного выхода при отклонении = 0 для пропорционального действия	REAL	-320 ... 320		0
MVTrackSw	Флаг слежения TRUE: ON FALSE: OFF	BOOL	-	---	FALSE

Название	Описание	Тип данных	Диапазон	Ед.изм.	По умолч.
MVTrackVal	Значение которое устанавливается на MV когда MV слежение включено	REAL	-320 ... 320	%	0
StopMV	Значение которое устанавливается на MV когда MV слежение остановлено	REAL			
ErrorMV	Значение которое устанавливается на MV при ошибке	REAL			
Alpha	Коэффициент α в фильтре задания Если равен 0 то фильтр отключен	REAL	0.00 ... 1.00	---	0.65
ATCalcGain	Коэффициент для автонастройки При большем значении предпочтение отдается устойчивости. При меньшем значении предпочтение отдается скорости реакции.	REAL	0.1 ... 10.0		1.0
ATHystrs	Гистерезис предела цикла.	REAL			% FS

Структура **_sINIT_SET_PARAMS**

Название	Описание	Тип данных	Диапазон	Ед.изм.	По умолч.
InitSetParams	Initial setting parameters.	_sINIT_SET_PARAMS	---	---	---

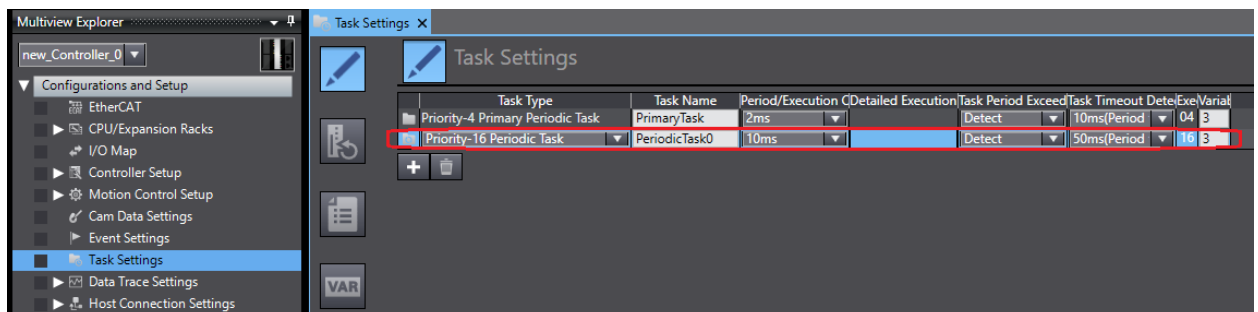
Название	Описание	Тип данных	Диапазон	Ед.изм.	По умолч.
SampTime	Период пересчета ПИД регулятора	TIME	T#0.0001 s ... #100.0000 s	s	T#0.1 s
RngLowLmt	Нижний предел PV и SP.	REAL	- 32000 ... 32000	---	0
RngUpLmt	Верхний предел PV и SP.	REAL			100
DirOpr	TRUE: прямое управление FALSE: инверсное управление	BOOL			FALSE

3. Тестовый пример в Sysmac Studio

Создадим новый проект в среде Sysmac Studio для контроллера. В конкретном примере используется контроллер NJ 101-1020.

Проект будет запускаться в режиме симуляции, так как в качестве объекта управления будет использоваться пользовательский ФБ, реализующий апериодическое звено 1-го порядка (имитация простого теплового объекта). Соответственно, ПИД регулятор будет замкнут на него.

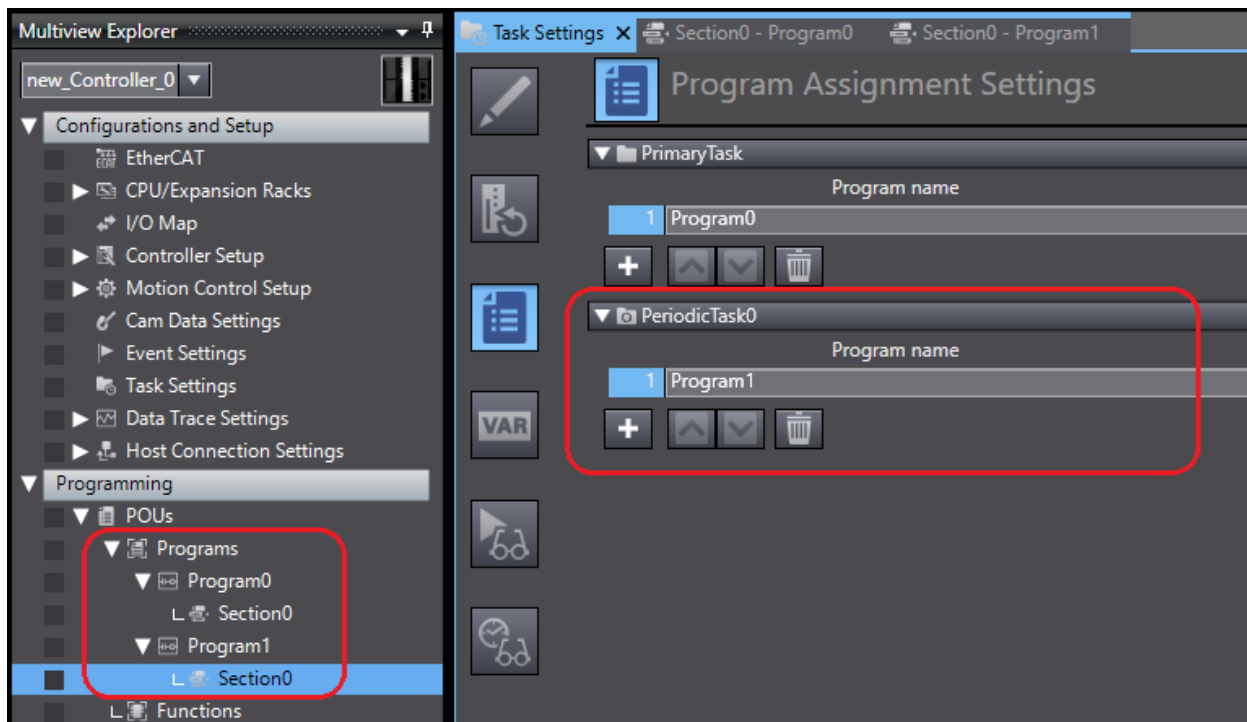
Откроем **Task Settings -> Task Settings** и добавим новую периодическую задачу **PeriodicTask0** с приоритетом 16 и циклом пересчета в 10 мс:



Проверить, что у всех программ выставлен признак **Initial Status: Run**

Предполагается, что в этой новой задаче у нас будет «крутиться» ПИД-регулятор.

Далее создадим новую программу Program1 в разделе Programming. И присвоим ее новой созданной ранее задаче **PeriodicTask0** в разделе **Task Settings -> Program Assignment Settings** :



ПИД регулятор должен чем-то управлять. Так как реального объекта у нас нет, то создадим функциональный блок, имитирующий объект управления.

Напишем функциональный блок на языке ST. Будем имитировать простой тепловой объект (типа нагреваемая емкость с водой) 1-го порядка.

Интерфейс входов и выходов ФБ Obj1:

Internals	Name	In/Out	Data Type	Edge	Initial Value	Retain	Constant
In/Out	in	Input	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>
Externals	out	Output	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>
	upr	Input	REAL	No Edge	5	<input type="checkbox"/>	<input type="checkbox"/>
	X	Output	REAL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>
	K	Input	real	No Edge	0.0	<input type="checkbox"/>	<input type="checkbox"/>
	T	Input	real	No Edge	0	<input type="checkbox"/>	<input type="checkbox"/>
	cycle	Input	real	No Edge	0	<input type="checkbox"/>	<input type="checkbox"/>

где upr – это управляющий вход объекта; X – это выход объекта (сигнал обратной связи, датчик); k, T – параметры объекта, его инерционность; cycle – это период пересчета задачи.

Код на языке ST блока Obj1:

Namespace - Using						
Internals	Name	Data Type	Initial Value	AT	Retain	Constant
In/Out						
Externals	prX	real			<input type="checkbox"/>	<input type="checkbox"/>
Externals	dtc	real			<input type="checkbox"/>	<input type="checkbox"/>

```

1 dtc:=cycle/1000.0;
2 X := prX + (-prX/T + K*upr/T)*dtc ;
3 prX:=X;

```

Пояснение данного кода приведено в **Приложении 1**.

Далее можем приступить к разработке основной программы. Её мы будем писать на языке LAD в программе **Program1**. (Программу Program0 однако нельзя оставлять пустой, нужно вбить хотя бы одну простейшую цепочку).

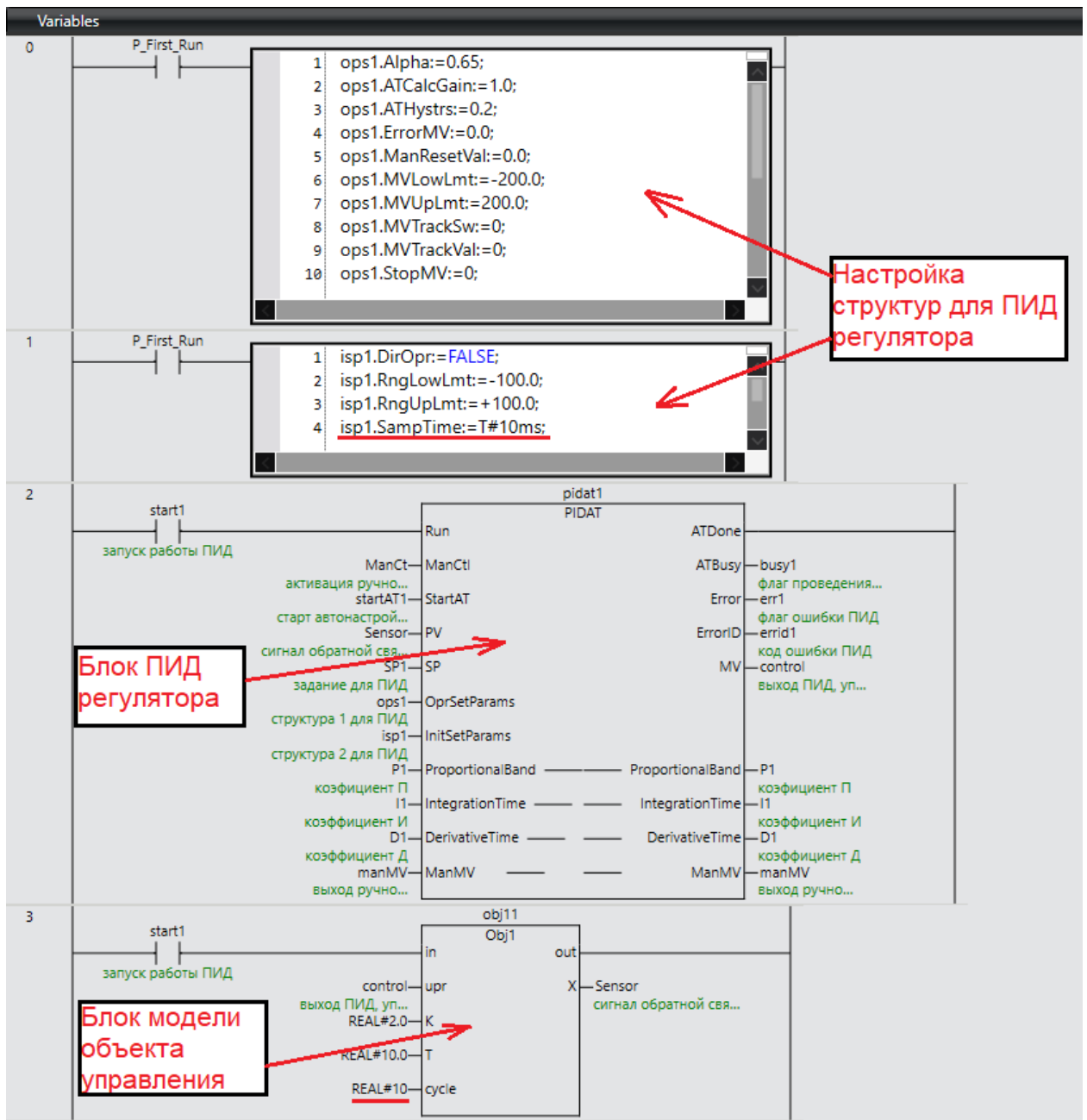
Объявим структуры и другие переменные:

Variables							
Namespace - Using							
Internals	Name	Data Type	Initial Value	AT	Retain	Constant	Comment
Externals	ops1	_sOPR_SET_PARAMS			<input type="checkbox"/>	<input type="checkbox"/>	структура 1 для ПИД
	isp1	_sINIT_SET_PARAMS			<input type="checkbox"/>	<input type="checkbox"/>	структура 2 для ПИД
	start1	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	запуск работы ПИД
	pidat1	PIDAT			<input type="checkbox"/>	<input type="checkbox"/>	экземпляр ФБ ПИД регулятора
	ManCt	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	активация ручного управления ПИД
	startAT1	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	старт автонастройки ПИД
	Sensor	REAL			<input type="checkbox"/>	<input type="checkbox"/>	сигнал обратной связи (датчик)
	SP1	REAL	10		<input type="checkbox"/>	<input type="checkbox"/>	задание для ПИД
	P1	REAL	100		<input type="checkbox"/>	<input type="checkbox"/>	коэффициент П
	I1	TIME	T#1s		<input type="checkbox"/>	<input type="checkbox"/>	коэффициент И
	D1	TIME	T#0S		<input type="checkbox"/>	<input type="checkbox"/>	коэффициент Д
	manMV	REAL			<input type="checkbox"/>	<input type="checkbox"/>	выход ручного управления
	busy1	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	флаг проведения автонастройки ПИД
	err1	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	флаг ошибки ПИД
	errid1	WORD			<input type="checkbox"/>	<input type="checkbox"/>	код ошибки ПИД
	control	REAL			<input type="checkbox"/>	<input type="checkbox"/>	выход ПИД, управляющий объектом сигнал
	obj11	Obj1			<input type="checkbox"/>	<input type="checkbox"/>	экземпляр ФБ модели объекта управления 1 порядка
	test_control	REAL			<input type="checkbox"/>	<input type="checkbox"/>	

Серым выделены экземпляры функциональных блоков, используемых в программе.

Также вы можете видеть предустановленные значения у некоторых переменных. Это задание (SP1=10), и три коэффициента регулятора (P1=100, I1=1 с, D1=0 с). Ограничимся ПИ-регулятором в данной тестовой программе.

Рассмотрим код тестовой программы:



Параметр `SampTime`, а также вход `cycle` у блока модели ОУ следует указать равным 10 мс, так как период пересчета задачи Periodic Task 0 равен именно этому значению.

Связка ПИД регулятора и объекта осуществляется посредством следующих сигналов:

- 1) Выход регулятора **control** приходит на вход объекта (вход **upr**);
- 2) Выход объекта **X** (переменная **Sensor**) приходит на вход ПИД регулятора **PV**.

В результате получается замкнутая система.

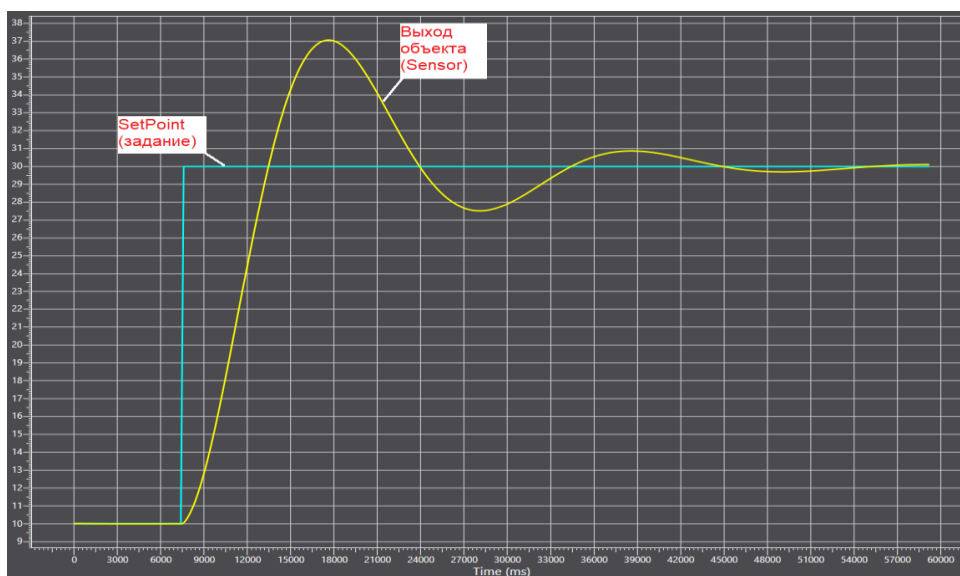
При запуске программы ПИД регулятор остановлен.

По включению флага **start1** связка ПИД регулятор + объект начинает работать. Регулятор выводит объект на задание в 10 усл.ед.

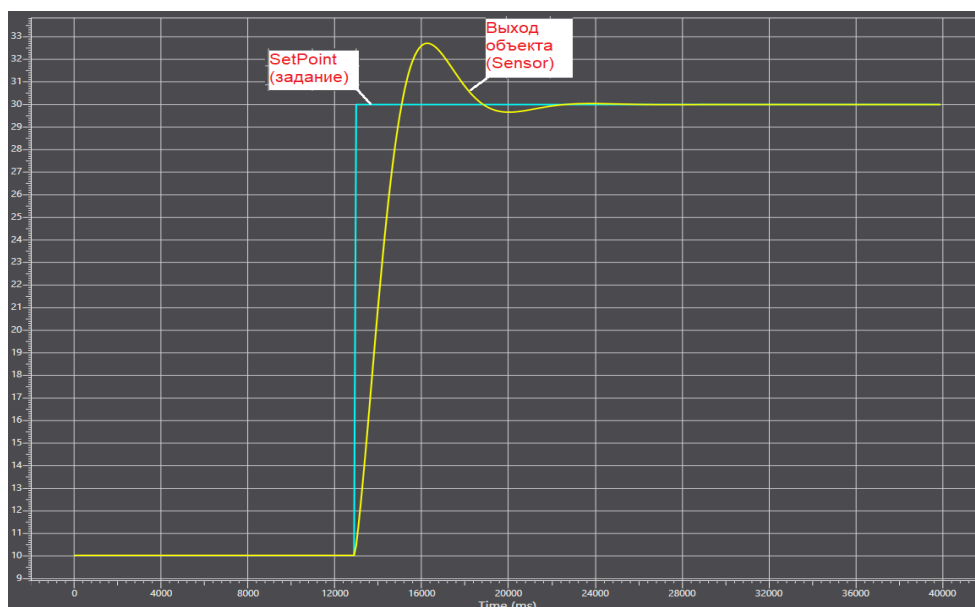
Требуется ввести новое значение задания SP и наблюдать как регулятор его обрабатывает.

Коэффициенты P1,I1,D1 возможно изменять прямо на ходу.

- 1) Переходный процесс при значениях параметров $P1=100$, $K1=1$, $D1=0$ и при скачке задания от 10 до 30 у.е. приведены на графике:



- 2) Переходный процесс при значениях параметров $P1=10$, $K1=1$, $D1=0$ и при скачке задания от 10 до 30 у.е. приведены на графике:



Таким образом, варьируя коэффициенты ПИД регулятора можно добиться оптимального переходного процесса при обработке задания.

При включении флага **StartAT1** начинается автонастройка регулятора. В результате коэффициенты P,I,D сами изменяются. Чтобы автонастройка прошла успешнее рекомендуется подать разные задания на регулятор и дождаться завершения переходного процесса. После автонастройки следует выключить флаг **StartAD1**.

Приложение 1. Пояснение блока моделирующего объект 1-го порядка

В данном коде мы имитируем объект, которым управляет ПИД-регулятор. Мы ограничились объектом 1-го порядка, представленным в непрерывном виде в виде передаточной функции:

$$W = \frac{k}{T*s+1} \quad (1)$$

где k – коэффициент усиления (определяет соотношение между амплитудой входного управляющего сигнала и выхода объекта, то есть зависит от типа сигнала датчика, который измеряет выход объекта);

T – постоянная времени (определяет инерционность объекта, то, как долго будет расти его выход при подачи управляющего входного импульса);

s – оператор Лапласа (для записи передаточной функции в частотной области). Такой моделью могут быть описаны простые тепловые объекты (чайник с водой). Записанную выше передаточную функцию можно представить в виде диф.уравнения:

$$T \frac{dx}{dt} + x = k * u \quad (2)$$

где u – управляющее воздействие на объект (куда подключается выход ПИД регулятора);

x – это регулируемый параметр объекта, его выход, например температура. Считаем что этот параметр измеряется датчиком и в качестве обратной связи сигнал подается на вход ПИД регулятора.

Уравнение (2) можно записать в разностном виде, если представить производную как конечную разность на малом интервале времени Δt :

$$\frac{dx}{dt} = \frac{x_i - x_{i-1}}{\Delta t}$$

Тогда уравнение (2) запишется в виде:

$$T \frac{x_i - x_{i-1}}{\Delta t} + x_{i-1} = k * u_i$$

Выразив из уравнения x_i получаем:

$$x_i = x_{i-1} + \left(-\frac{x_{i-1}}{T} + k \frac{u_i}{T} \right) * \Delta t$$

Получили рекуррентное уравнение, которое может быть запрограммировано в коде программы (см. строку 2 в тексте функционального блока).

Интервал времени Δt - это интервал пересчета программы ПЛК, где исполняется ПИД регулятор.