

# ПИД-РЕГУЛЯТОР

## ВВЕДЕНИЕ

База знаний (внешние документы)

- СИСТЕМЫ РЕГУЛИРОВАНИЯ.pdf : стр.47-72

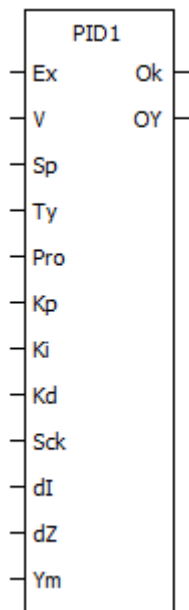
Примеры реализации алгоритма (см. далее)

- [С ТАКТИРОВАНИЕМ ОТ ВРЕШНЕГО ДИСКРЕТНОГО СИГНАЛА](#)
- [С ТАКТИРОВАНИЕМ ОТ СОБСТВЕННОГО ПРОГРАММНОГО ТАЙМЕРА](#)

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ ВНЕШНЕГО ДИСКРЕТНОГО СИГНАЛА

### функциональный блок



#### Входы:

Ex — разрешение работы блока [BOOL]

= FALSE (по умолчанию)

= TRUE

X — контролируемый параметр [REAL]

= 0.0 (по умолчанию)

Sp — уставка (в единицах значения X) [REAL]

= 0.0 (по умолчанию)

Ty — тип регулятора [BYTE]

= 0 — ПИД (по умолчанию)

= 1 — ПИ

= 2 — ПД

= 3 — П

Pro — процесс изменения управляющего сигнала (Y) [BYTE]

= 0 — увеличение (например, нагрев) (по умолчанию)

= 1 — уменьшение (например, охлаждение)

Kp — коэффициент пропорции [REAL]

= 0.0 (по умолчанию)

Ki — коэффициент интеграла [REAL]

= 0.0 (по умолчанию)

Kd — коэффициент дифференциала [REAL]

= 0.0 (по умолчанию)

Sck — внешний дискретный сигнал тактирования регулятора [BOOL]

= FALSE (по умолчанию) — не выполнять расчет составляющих регулятора

= TRUE — выполнять расчет составляющих регулятора

dI — зона интегральной составляющей (в единицах значения X) [REAL]

= 0.0 (по умолчанию)

dZ — зона нечувствительности (в единицах значения X) [REAL]

= 0.0 (по умолчанию)

dE — допуск контролируемого параметра ( $\pm X$ ) для диагностики контура управления [REAL]

= 0.0 (по умолчанию)

dETm — время диагностики контура управления (мс) [DWORD]

= 5000 (по умолчанию)

Ym — тип выхода регулятора [BYTE]

= 0 - униполярный (от 0.0 до 100.0 %) (по умолчанию)

= 1 — биполярный (от -100.0 до 100.0 %).

#### Выходы:

Ok — статус работы регулятора [BYTE]

= 0 — останов

= 1 — работа

= 2 — работа в dZ

= 3 — работает в dI

= 4 — обрыв контура управления

= 5 — заклинивание контура управление

OY — управляющий сигнал (%) [REAL]

= 0.0 (по умолчанию)

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ ВНЕШНЕГО ДИСКРЕТНОГО СИГНАЛА

реализация на языке ST (стандарт IEC-61131-3)

таблица переменных

#	Имя	Класс	Тип	Исходное значение	Квалификатор	Описание
1	Ex	Вход	BOOL	FALSE		execution allow
2	V	Вход	REAL	0.0		source value
3	Sp	Вход	REAL	0.0		setpoint
4	Ty	Вход	BYTE	0		type of controller
5	Pro	Вход	BYTE	0		type of process
6	Kp	Вход	REAL	0.0		P-factor
7	Ki	Вход	REAL	0.0		I-factor
8	Kd	Вход	REAL	0.0		D-factor
9	Sck	Вход	BOOL	FALSE		external command to start regulator cycle
10	dI	Вход	REAL	0.0		integral zone
11	dZ	Вход	REAL	0.0		dead zone
12	Ym	Вход	BYTE	0		type of OY
13	Ok	Выход	BYTE	0		result code
14	OY	Выход	REAL	0.0		regulator value (%)
15	Er	Локальный	REAL	0.0		error delta
16	ErAbs	Локальный	REAL	0.0		error delta (absolute)
17	ErPrev	Локальный	REAL	0.0		error delta (previous cycle)
18	ErDiff	Локальный	REAL	0.0		error delta (diff)
19	Ppart	Локальный	REAL	0.0		P-part
20	Ipart	Локальный	REAL	0.0		I-part
21	Dpart	Локальный	REAL	0.0		D-part
22	SckSet	Локальный	BOOL	FALSE		external start (set)
23	ErrorSet	Локальный	BOOL	FALSE		e-zone (set)

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ ВНЕШНЕГО ДИСКРЕТНОГО СИГНАЛА

реализация на языке ST (стандарт IEC-61131-3)

КОД

```
(* reset Sck *)
IF Sck = FALSE THEN
  SckSet:= FALSE;
END_IF;

(* test execution allow *)
IF Ex = TRUE THEN

  (* test regulator type *)
  IF Ty <= 3 THEN

    (* test start *)
    IF Sck = TRUE AND SckSet = FALSE AND ErrorSet = FALSE THEN

      SckSet:= TRUE;

      (* work *)
      Ok:= 0;

      Er      := (Sp-V);
      ErDiff:= (Er-ErPrev);
      ErPrev:= Er;
      ErAbs  := ABS(IN:=Er);

      (* test d-zone *)
      IF ErAbs >= dZ THEN

        (* Ty: 0-PID, 1-PI, 2-PD, 3-P *)
        Ppart:= (Kp*Er);

        (* Ty: 0-PID, 1-PI *)
        IF (Ty = 0 OR Ty = 1) AND (ErAbs < dI OR dI = 0.0) AND Ki > 0.0 THEN
          (* i-zone *)
          Ok:= 3;
          Ipart:= (Ipart+Er);
          IF Ipart >= Ki THEN
            Ipart:= Ki;
          END_IF;
          Ipart:= ((1.0/Ki)*Ipart);
        ELSE
          Ipart:= 0.0;
        END_IF;

        (*0-PID, 2-PD*)
        IF (Ty = 0 OR Ty = 2) THEN
          Dpart:= (Kd*ErDiff);
        ELSE
          Dpart:= 0.0;
        END_IF;

        OY:= (Ppart+Ipart+Dpart)*100.0;

      ELSE

        (* d-zone *)
        Ok:= 2;
      END_IF;
    (* end of test d-zone *)

  END_IF;
(* end of test start *)

END_IF;
```

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ ВНЕШНЕГО ДИСКРЕТНОГО СИГНАЛА

реализация на языке ST (стандарт IEC-61131-3)

КОД

```
ELSE
  (* incorrect regulator type *)
  Ok:= 4;
  ErrorSet:= TRUE;
END_IF;
(* end of test regulator type *)

ELSE
  (* off *)
  Ok:= 1;
END_IF;
(* end of Ex *)

(* off *)
IF Ok = 1 THEN
  OY      := 0.0;
  SckSet  := FALSE;
  ErrorSet:= FALSE;
  Ppart   := 0.0;
  Ipart   := 0.0;
  Dpart   := 0.0;
END_IF;

(* error of regulator type *)
IF ErrorSet = TRUE THEN
  OY      := 0.0;
END_IF;

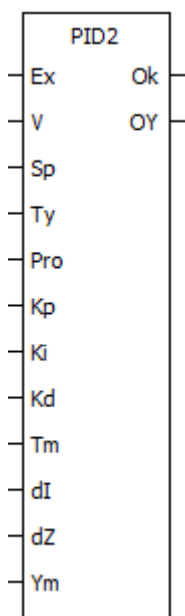
(* scale OY by type of process *)
IF Pro = 1 THEN
  (* decreasing *)
  OY:= (100.0-OY);
END_IF;

(* scale OY by type of output *)
IF Ym = 1 THEN
  (* bipolar *)
  OY:= LIMIT(MN:=-100.0, IN:=OY, MX:=100.0);
ELSE
  (* unipolar *)
  OY:= LIMIT(MN:=0.0, IN:=OY, MX:=100.0);
END_IF;
```

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ СОБСТВЕННОГО ПРОГРАММНОГО ТАЙМЕРА

### функциональный блок



#### Входы:

Ex — разрешение работы блока [BOOL]  
= FALSE (по умолчанию)  
= TRUE

X — контролируемый параметр [REAL]  
= 0.0 (по умолчанию)

Sp — уставка (в единицах значения X) [REAL]  
= 0.0 (по умолчанию)

Ty — тип регулятора [BYTE]  
= 0 – ПИД (по умолчанию)  
= 1 – ПИ  
= 2 – ПД  
= 3 – П

Pro – процесс изменения управляющего сигнала (Y) [BYTE]  
= 0 – увеличение (например, нагрев) (по умолчанию)  
= 1 – уменьшение (например, охлаждение)

Kp – коэффициент пропорции [REAL]  
= 0.0 (по умолчанию)

Ki – коэффициент интеграла [REAL]  
= 0.0 (по умолчанию)

Kd – коэффициент дифференциала [REAL]  
= 0.0 (по умолчанию)

Tm — время задержки выполнения алгоритма регулятора (мсек) [BYTE]  
= 1000 (по умолчанию)

dI — зона интегральной составляющей (в единицах значения X) [REAL]  
= 0.0 (по умолчанию)

dZ — зона нечувствительности (в единицах значения X) [REAL]  
= 0.0 (по умолчанию)

dE — допуск контролируемого параметра ( $\pm X$ ) для диагностики контура управления [REAL]  
= 0.0 (по умолчанию)

dETm — время диагностики контура управления (мс) [DWORD]  
= 5000 (по умолчанию)

Ym — тип выхода регулятора [BYTE]  
= 0 - униполярный (от 0.0 до 100.0 %) (по умолчанию)  
= 1 – биполярный (от -100.0 до 100.0 %).

#### Выходы:

Ok – статус работы регулятора [BYTE]  
= 0 – останов

= 1 – работа

= 2 – работа в dZ

= 3 – работает в dI

= 4 – обрыв контура управления

= 5 – заклинивание контура управления

OY – управляющий сигнал (%) [REAL]  
= 0.0 (по умолчанию)

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ СОБСТВЕННОГО ПРОГРАММНОГО ТАЙМЕРА

реализация на языке ST (стандарт IEC-61131-3)

таблица переменных

#	Имя	Класс	Тип	Исходное значение	Квалификатор	Описание
1	Ex	Вход	BOOL	FALSE		execution allow
2	V	Вход	REAL	0.0		source value
3	Sp	Вход	REAL	0.0		setpoint
4	Ty	Вход	BYTE	0		type of controller
5	Pro	Вход	BYTE	0		type of process
6	Kp	Вход	REAL	0.0		P-factor
7	Ki	Вход	REAL	0.0		I-factor
8	Kd	Вход	REAL	0.0		D-factor
9	Tm	Вход	DWORD	1000		cycle time (ms)
10	dI	Вход	REAL	0.0		integral zone
11	dZ	Вход	REAL	0.0		dead zone
12	Ym	Вход	BYTE	0		type of OY
13	Ok	Выход	BYTE	0		result code
14	OY	Выход	REAL	0.0		regulator value (%)
15	TON_Work	Локальный	TON			
16	TON_Work_IN	Локальный	BOOL	FALSE		
17	TON_Work_TM	Локальный	TIME	T#0ms		
18	PID1_1	Локальный	PID1			

# ПИД-РЕГУЛЯТОР

## С ТАКТИРОВАНИЕМ ОТ СОБСТВЕННОГО ПРОГРАММНОГО ТАЙМЕРА

реализация на языке ST (стандарт IEC-61131-3)

### КОД

```
PID1_1(Ex:=Ex, V:=V, Sp:=Sp, Ty:=Ty, Pro:=Pro,  
      Kp:=Kp, Ki:=Ki, Kd:=Kd, Sck:=TON_Work.Q, dI:=dI, dZ:=dZ, Ym:=Ym);
```

```
Ok:= PID1_1.Ok;
```

```
OY:= PID1_1.OY;
```

```
IF Ex = TRUE AND Tm > 0 THEN
```

```
    TON_Work_TM:= REAL_TO_TIME(DWORD_TO_REAL(Tm)/1000.0);
```

```
    TON_Work_IN:= TRUE;
```

```
ELSE
```

```
    TON_Work_IN:= FALSE;
```

```
END_IF;
```

```
IF TON_Work.Q = TRUE THEN
```

```
    TON_Work_IN:= FALSE;
```

```
END_IF;
```

```
TON_Work(IN:=TON_Work_IN, PT:=TON_Work_TM);
```