

PID-functions

Set of helper functions to initiate and update a PID control loop

Description

```
PID_new( pid[], Float:set, Float:out ) // set up new PID
```

```
PID_dir( pid[], direction ) // set direction of control
```

```
PID_set( pid[], Float:set ) // adjust the target value
```

```
PID_tune( pid[], Float:Kp, Float:Ki, Float:Kd ) // set the P, I, D parameters
```

```
PID_limits( pid[], Float:min, Float:max ) // set the control range
```

```
PID_update( pid[], Float:in ) // update PID with feedback, and return new control value
```

Parameters

set	setpoint / target
out	Initial output value
direction	1 (normal) or -1 (reverse)
Kp	Proportional component
Ki	Integral response component
Kd	Derivative response component
min	Smallest output value allowed
max	Largest output value allowed

Return value

The PID_update function returns the calculated output value.

Example usage

```
new p[PID]; // create a PID

main()
{
    // Initialize the PID with target and no output
```

```
PID_new(p, 50.0, 0.0);

// Limit output to +/- 10
PID_limits(p, -10.0, 10.0);

// Set the tuning parameters
PID_tune(p, 0.2, 0.04, 0.01);

// Set update speed to 100ms
SetTickInterval( 100 );
}

@Tick(uptime)
{
    new Float:psi;
    new Float:v;

    // Read and scale a sensor input (0-100psi from 4-20mA sensor)
    psi = 100.0 * ((GetInputValue(1, INVAL_RAW)-4000)/16000);

    // Update the PID with the feedback value
    v = PID_update(p, psi);

    // Apply to output
    SetOutput(4, fround(v+50));
}
```

NOTE: The values used in the above example are not from a real world setup. Please make sure you understand how PID works and how to correctly tune the variables if you use any of the PID functions.

From:

<https://doc.eze.io/> - ezeio documentation

Permanent link:

https://doc.eze.io/ezeio2/scriptref/pid_new

Last update: **2022-03-30 23:44**

