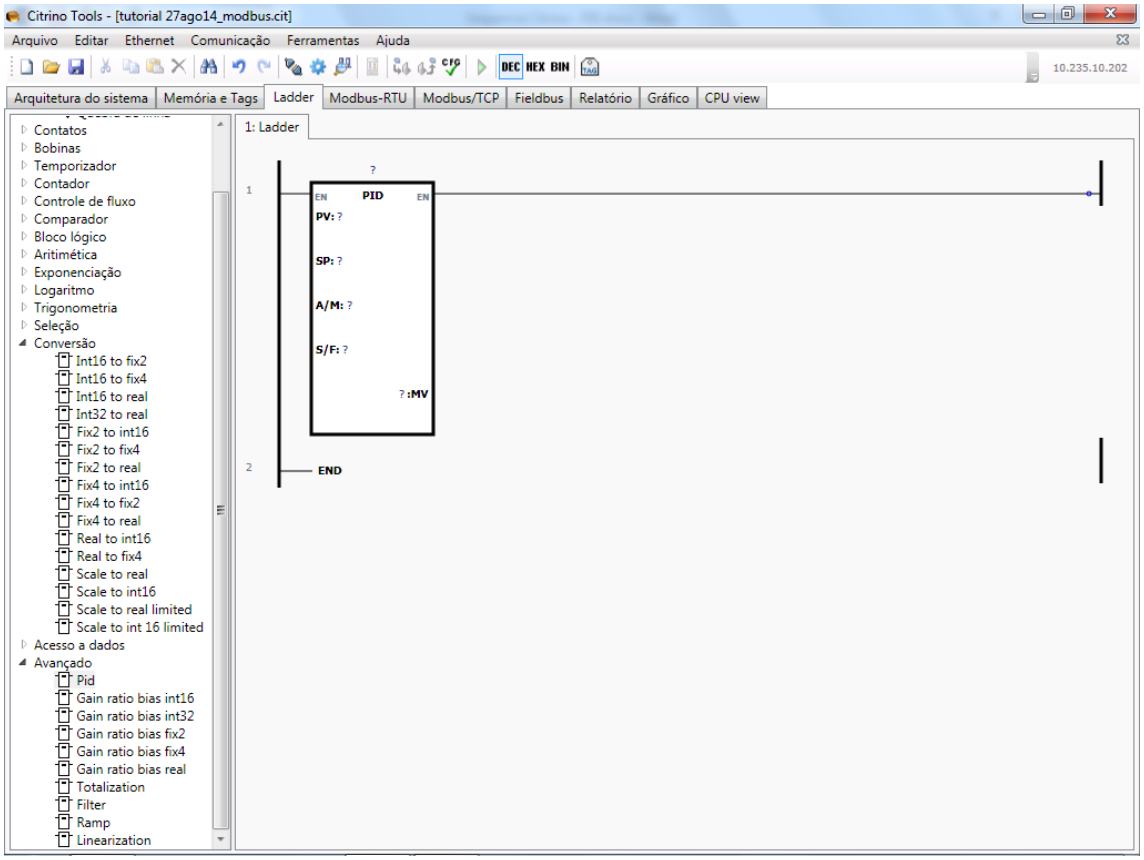
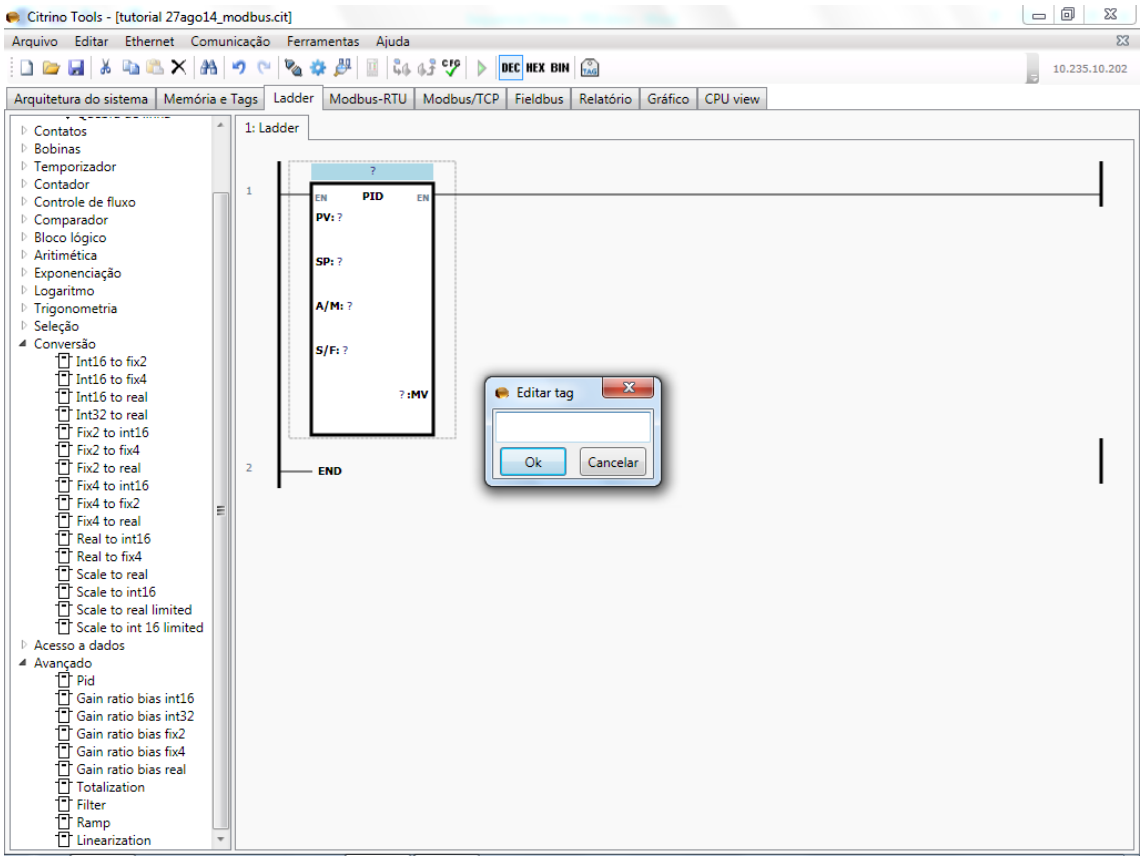
Tutorial experimento 1: Configuração de controlador PID no CLP Citrino

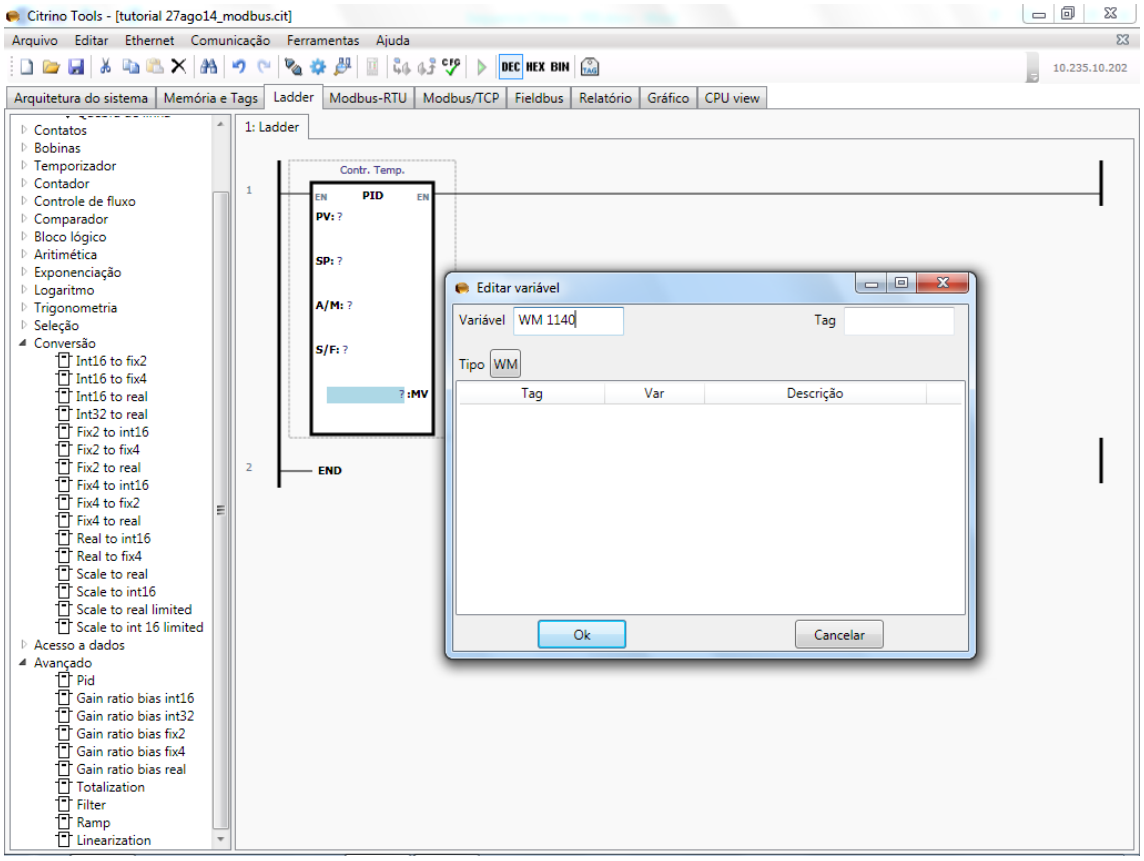
Siga as instruções adiante para inserir um bloco PID na lógica ladder. Dentro da pasta “Avançado”, localize o bloco PID e insira-o na lógica.



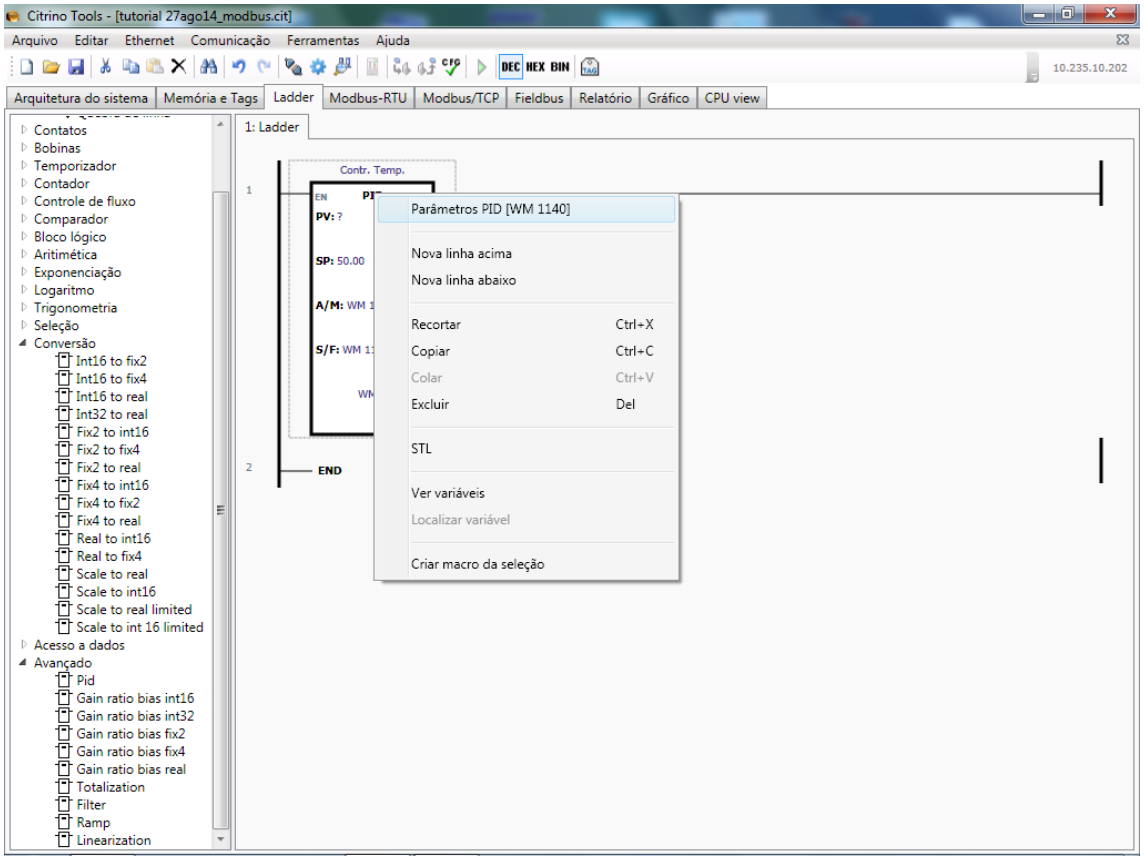
Clique com o botão direito no título do bloco defina um TAG para este controlador.



Clique com o botão direito no “?” relacionado ao MV e defina um endereço para alocar os parâmetros do bloco PID a partir deste endereço.

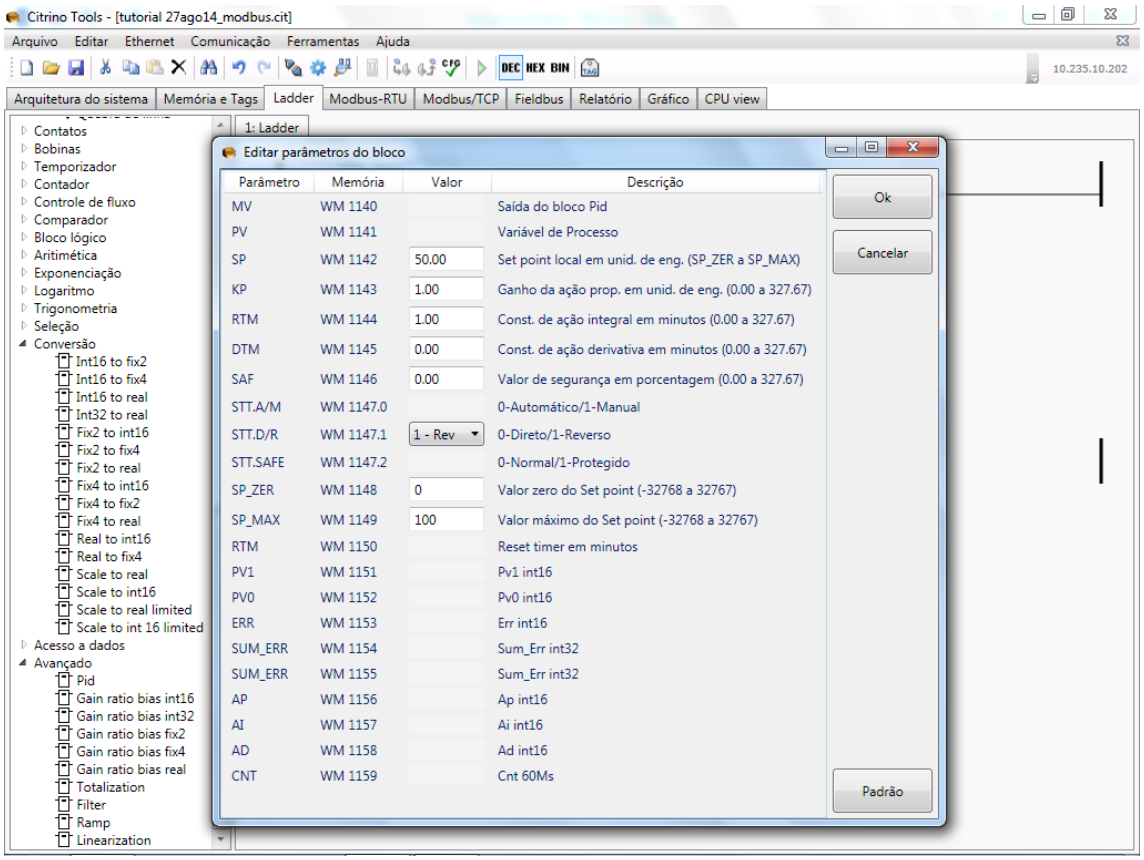


Clique com o botão direito sobre o bloco PID e selecione a opção Parâmetros PID



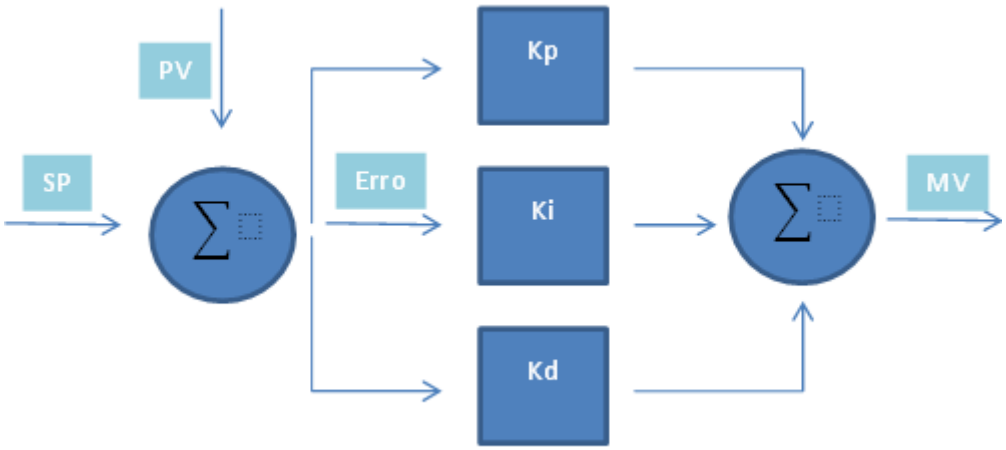
Verifique na janela apresentada a seguir que o conjunto de parâmetros internos do bloco PID foi alocado em 20 posições de memória WM (de 16bits) a partir do endereço WM definido para a MV.

Obs: Deixar sempre habilitado em modo “auto” e safe em “off” na parte externa do bloco PID.



Entre os parâmetros, estão aqueles que definem a sintonia do controlador: termos proporcional (P), integral (I) e derivativo (D), além do Set Point (SP).

Na prática o controlador P.I.D. do Citrino aplica em paralelo os termos P (Kp), I (Ki) e D (Kd) ao sinal de erro, semelhante ao diagrama de blocos a seguir:



As variáveis Kp, Ki e Kd na figura são definidas na memória do Citrino com inteiras de 16bits (WM) denominadas KP, RTM e DTM respectivamente, pois o bloco PID só pode ser posicionado na região de variáveis WM por definição. Como todas elas são de 16 bits (inteiras), os seus valores são limitados pelo software da seguinte maneira:

Termo proporcional KP = -32768 a 32767

Termo integrador RTM = 0 a 32767

Termo derivativo DTM = 0 a 32767

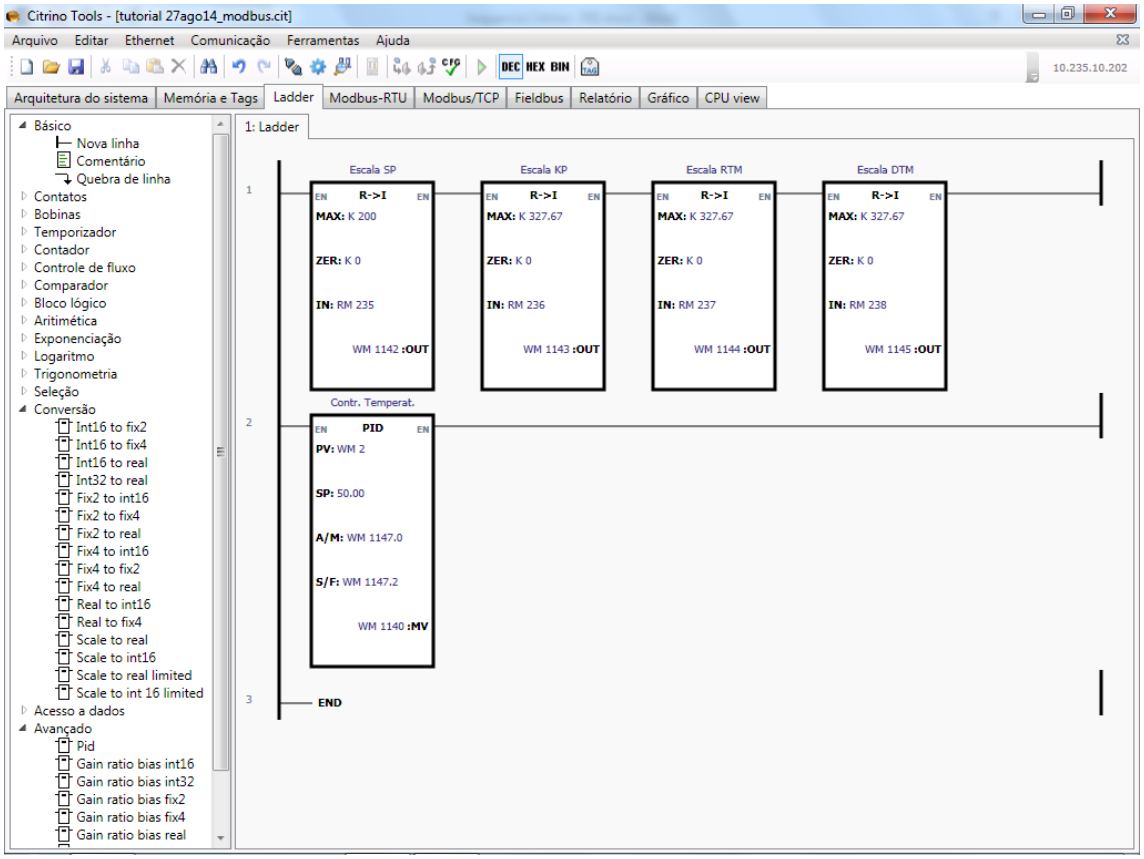
Porém, sabemos que para se fazer boas sintonias, é necessário que estes valores tenham pequenas granularidades e isto só é possível com pequenos acréscimos nestas constantes, como por exemplo alterar um KP de 1.00 para 1.01. Desta maneira, é preciso atentar para o seguinte aspecto de operação do controlador PID: o bloco PID "enxerga" estas variáveis como ponto fixo de duas casas decimais em minutos, ficando da seguinte maneira:

KP = -327.68 a 327.67

RTM = 0 a 327.67 (minutos)

DTM = 0 a 327.67 (minutos)

Desta forma, caso se necessite escrever os valores diretamente em número de ponto flutuante, deve-se realizar uma conversão de tipos entre Real e Inteiro. Isto pode ser realizado como no código a seguir:



Utilize os blocos "Real to Int16" dentro da opção "Conversão" para converter, por exemplo os números reais em RM236, 237 e 238 nos números inteiros em WM1143, 1144, 1145 respectivamente, WMs estes coincidentes com os WMs dos termos de sintonia do PID.

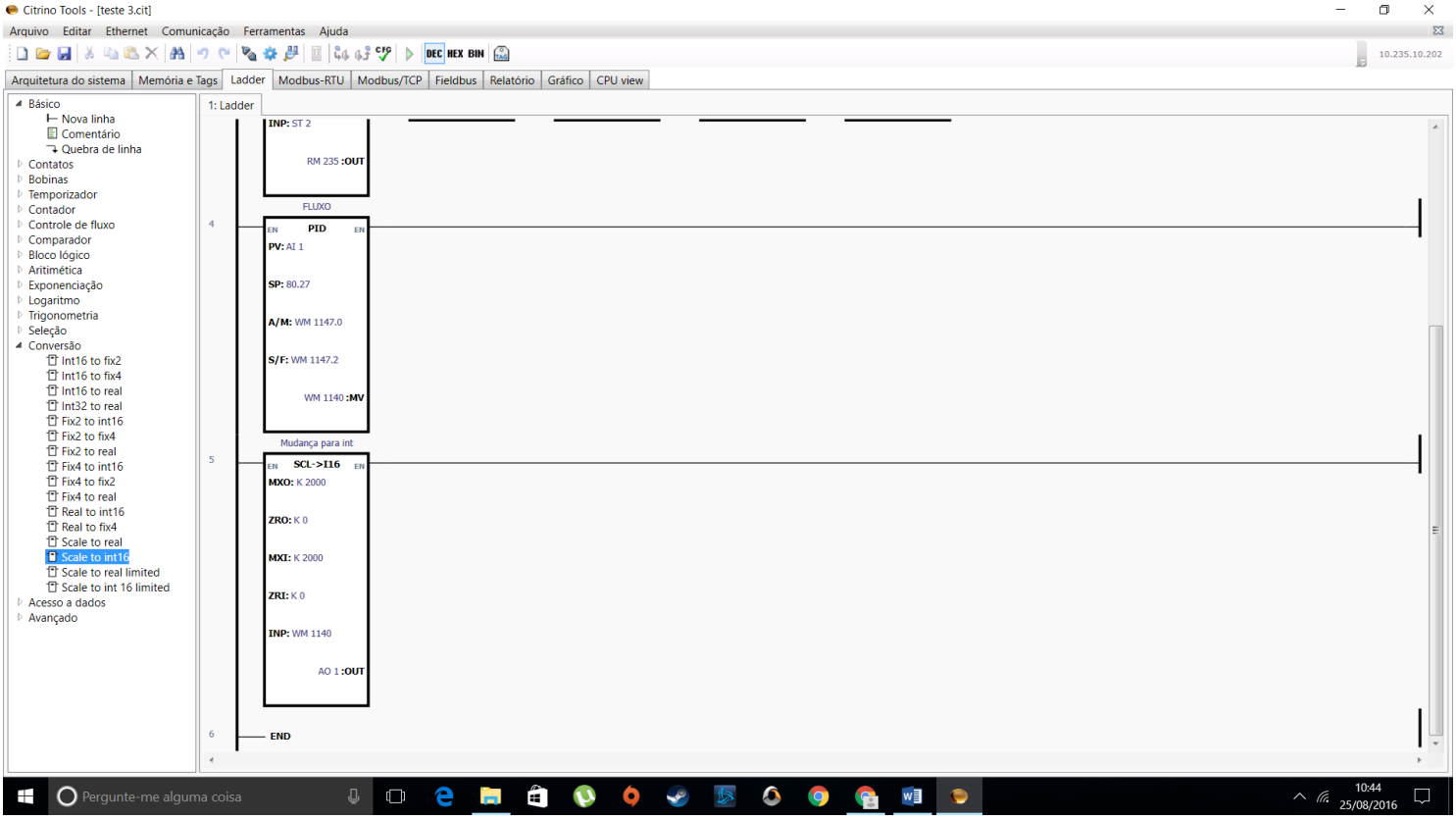
Verifique que neste exemplo, o fator de escala está 1 para 1 (o número 1 de ponto flutuante equivale a 1 também em inteiro, pois as escalas estão de 0 a 327.67).

A saída do bloco PID será um número na faixa de 0 a 32000, na variável MV.

A entrada da variável de processo (PV) pode ser definida pelo usuário como um endereço qualquer na região das variáveis WM com valores de 0 a 32000 (0 a 100% na escala do SP: SP_Zero e SP_Max). Use desta forma um bloco de conversão R -> I idêntico ao utilizado para o SP, com os mesmos valores max e min.

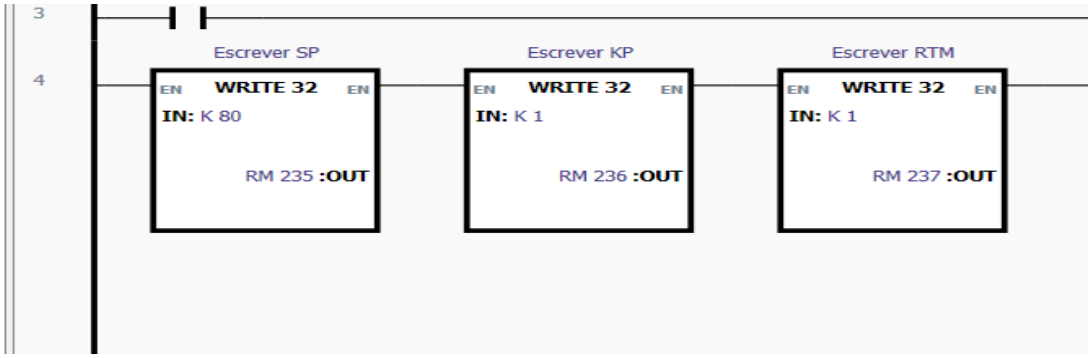
Por fim, adicione um bloco que coloque a saída dentro da escala desejada e ao mesmo tempo converta em um inteiro de 16 bits para a saída analógica desejada.

Como a imagem abaixo:



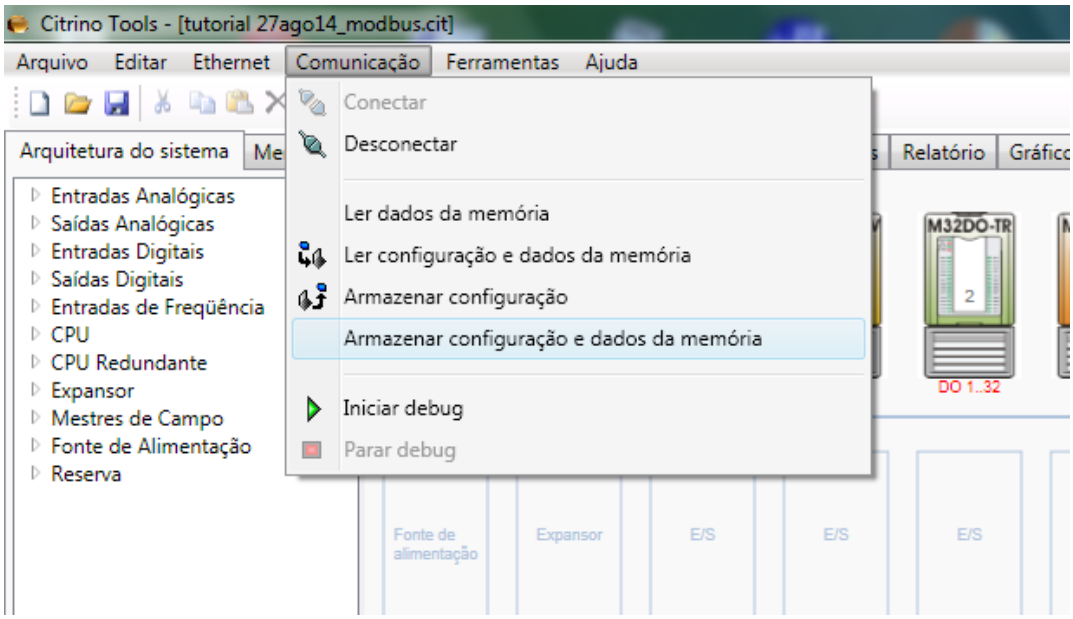
Para escrever nos parâmetros (SP,KP, RTM e DTM) do bloco PID, deve-se inserir blocos “write32” para cada um deles.

Em “IN” se deve inserir a entrada do ganho no parâmetro a ser modificado, utilizando o tipo de variável “KR” e na saída “OUT” o mesmo tipo e valor de memória onde irá converter e limitar a faixa de cada parâmetro (como descrito anteriormente), da seguinte maneira:



Muito importante:

Sempre que for armazenar uma configuração no CLP, utilizar o menu Comunicação -> Armazenar configuração e dados da memória.

<u>Não</u> utilizar o botão de atalho, pois por ele não se armazenam os dados da memória.

Veja imagem adiante: