



## Configurações diversas utilizadas na programação do CLP

The screenshot displays the Haiwell V2.2.5 PLC programming software interface. The main window shows a ladder logic program for 'PGB:Motor control'. It consists of three networks: Network 1 (start pulse), Network 2 (x0 start pulse), and Network 3 (motor control). Network 1 uses a TON timer (T2) with a set value of V1002. Network 2 uses a TP timer (T1) with a set value of V1001. Network 3 shows the motor control logic with outputs Y1 (back), Y2 (output), Y3 (stop), Y4 (fault), and Y5 (ready). The software interface includes a project manager on the left, a toolbar at the top, and a status bar at the bottom. A physical Haiwell PLC unit is shown in the bottom right corner, featuring a terminal block with inputs X0-X7 and outputs Y0-Y7, a power supply section, and a display showing 'T16S2P'.

# Curso de automação industrial utilizando o CLP Haiwell

AULA 8 | CONFIGURAÇÕES DIVERSAS

# Conteúdo da aula

8

Nesta aula iremos treinar diversos tópicos de configuração dos CLPs Haiwell que serão úteis no dia-a-dia dos programadores que estão acompanhando o curso.

O conteúdo desta aula também tem como objetivo preparar os alunos para a avaliação on-line para a obtenção do certificado de participação no treinamento CURSO DE AUTOMAÇÃO UTILIZANDO O CLP HAIWELL.

Para acompanhar esta aula é necessário ter concluído todas as anteriores.

Siga passo-a-passo a aula 8 para estar preparado para a avaliação.

Veja ao lado os assuntos desta aula.

- ▶ Iniciar um novo projeto
- ▶ Definição da área de memória retentiva
- ▶ Criação de blocos de programa
- ▶ Encriptação de projetos
- ▶ Criação de tabelas de inicialização de dados
- ▶ Adição de blocos matemáticos de função
- ▶ Instruções básicas de temporização
- ▶ Simulação do programa
- ▶ Monitoração do relógio de tempo real

# Crie um novo projeto

- ▶ Crie um novo projeto utilizando o modelo T16S0P-e e dê ao mesmo o nome “Operação básica Haiwell”
- ▶ Altere a área de dados retentivos da seguinte forma:
  - ▶ Operandos V retentivos iniciando em V500 e tamanho de 3000 registros
  - ▶ Operandos M retentivos iniciando em M1000 e tamanho de 2000 registros

A janela de criação de **New Project** deve ficar como a seguir



New project

PLC Series: T Series CPU Type: T16S0T/P(-e)

Auto save: 6 Minute

T16S0T/P(-e) (V0-V14847 M0-M12287 T0-T1023 C0-C255 S0-S2047)  
CPU module 8\*DI 8\*DO transistor DC24V power supply 2 channel 200KHz pulse input 2 channel 200KHz pulse output 2 communication ports support 7 extension modules

Power-off preservation (V500-V3499 M1000-M2999 T96-T127 C64-C127 S156-S255 )

	Start component	Length		Start component	Length	
V	500	3000		T	96	Clear
M	1000	2000		C	64	Default
S	156	100				

Project name: Operação básica Haiwell

User name:

Designer: Version:

Company:

Password: Confirm password:

Date created: 31/01/2018 17:33:30 Modified:

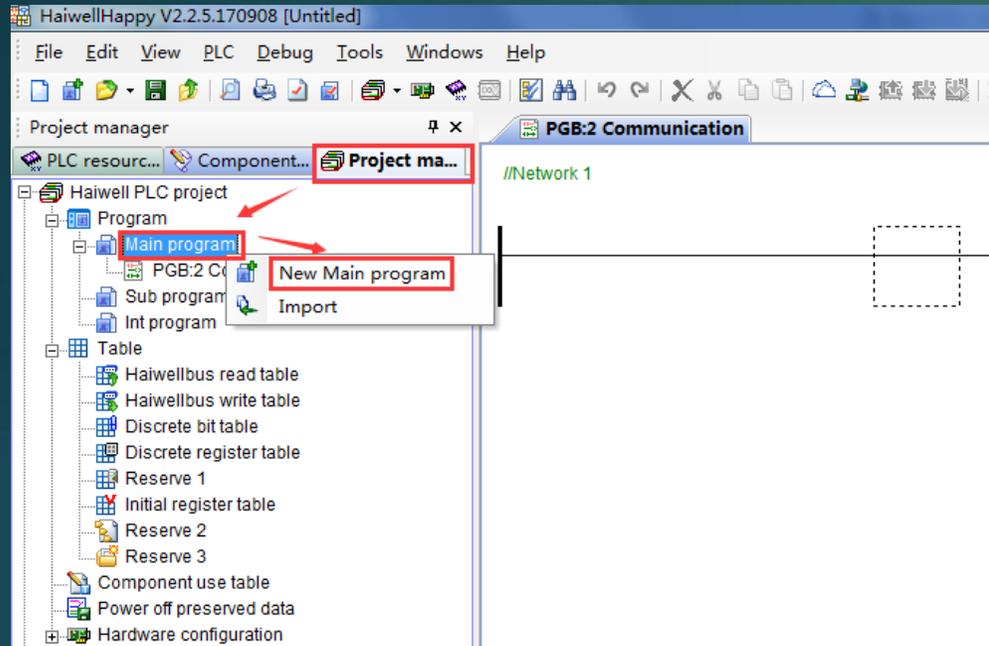
Comments:

OK Cancel

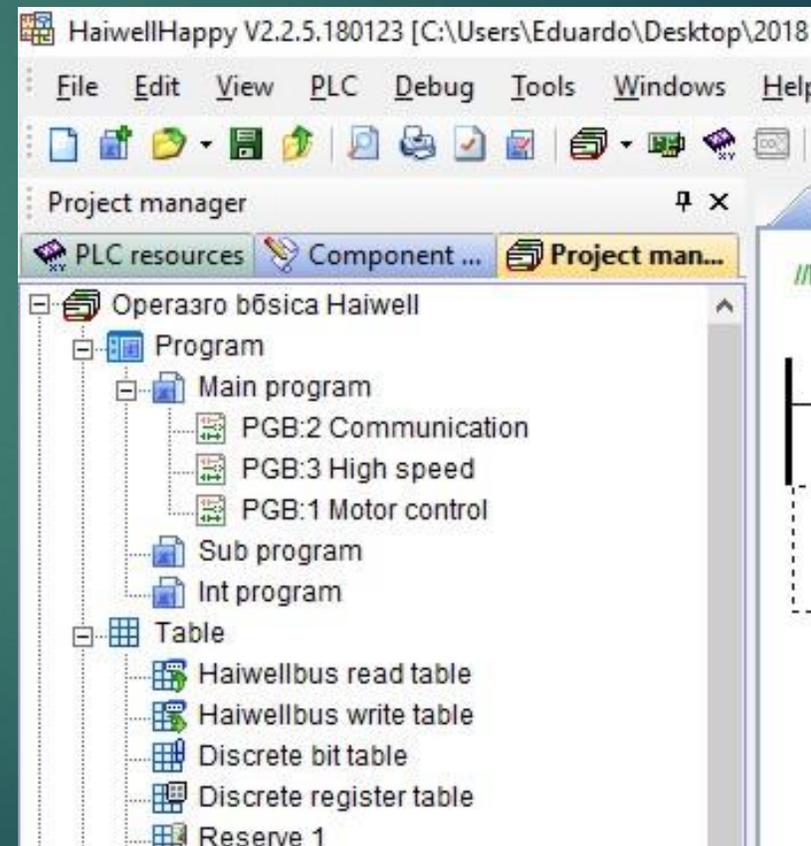
# Crie blocos de programa

- ▶ Crie três blocos de programa principais:
  - ▶ Bloco 2: 2 Communication
  - ▶ Bloco 3: 3 High speed
  - ▶ Bloco 1: 1 Motor control
- ▶ Observe que os blocos estão sendo criados propositalmente fora de sequência
- ▶ Após criar os três blocos, reordene os mesmos de forma a que o bloco “1 Motor control” seja o primeiro na sequência de processamento

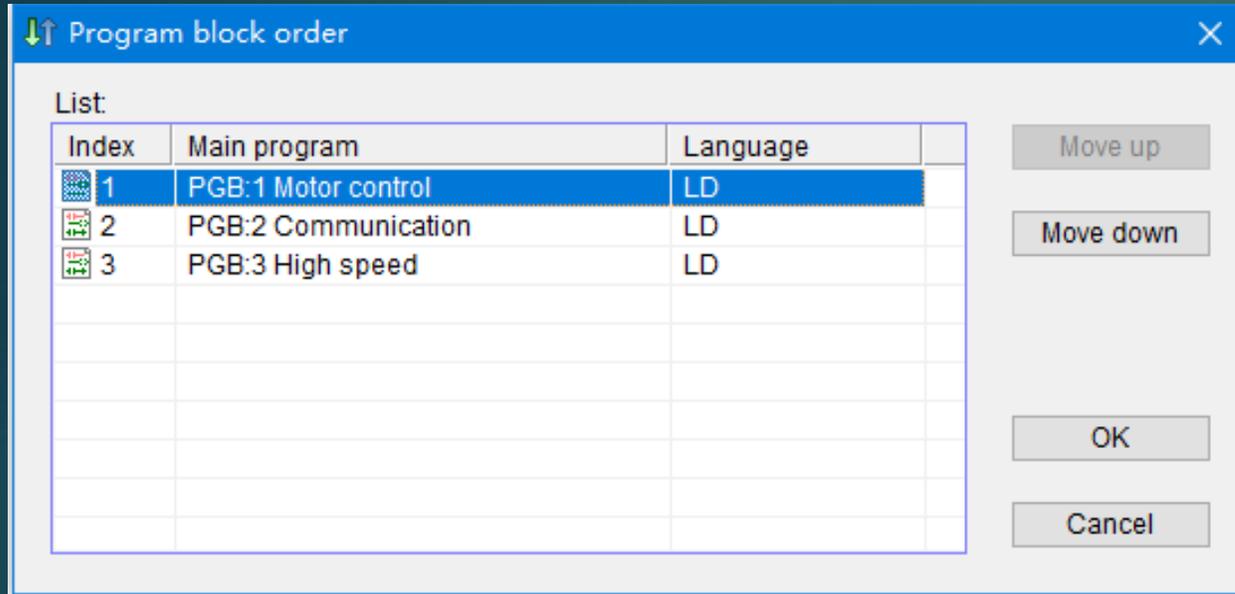
- ▶ Clique em **Main Program** para criar os blocos



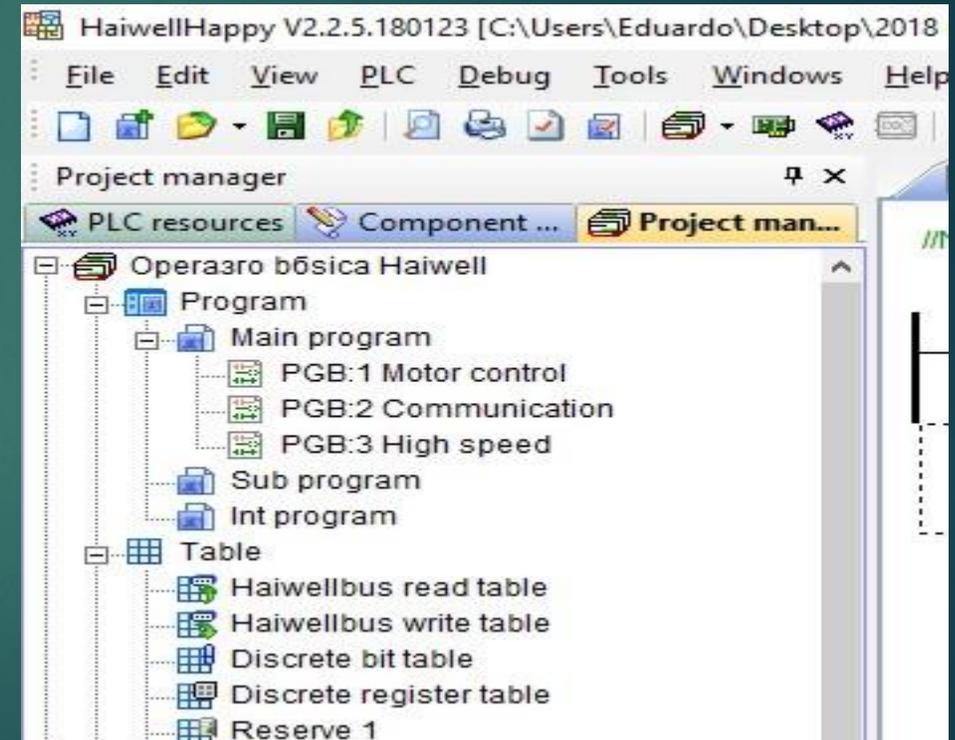
- ▶ Os blocos foram criados na sequência abaixo



- ▶ Clique em **Debug** para reordenar os blocos



- ▶ Depois de reordenados os blocos ficam assim organizados



# Proteja um bloco com senha

8

- ▶ Proteja o bloco de programa “1 Motor control” com a senha “hello”
  - ▶ Clique como botão direito sobre o bloco e selecione **Program property**
  - ▶ No campo **Password** insira a senha e confirme no campo **Confirm password**
- ▶ Salve o projeto, feche o HaiwellHappy, reabra o HaiwellHappy e recarregue o projeto. Tente abrir o bloco “1 Motor control. Observe que agora a senha é exigida

Main program Project properties

Block name: 1 Motor control

Type

- Main program
- Sub program
- Int program

Language

- LD
- FBD
- IL
- SFC
- ST

Comments

Password: \*\*\*\*\*

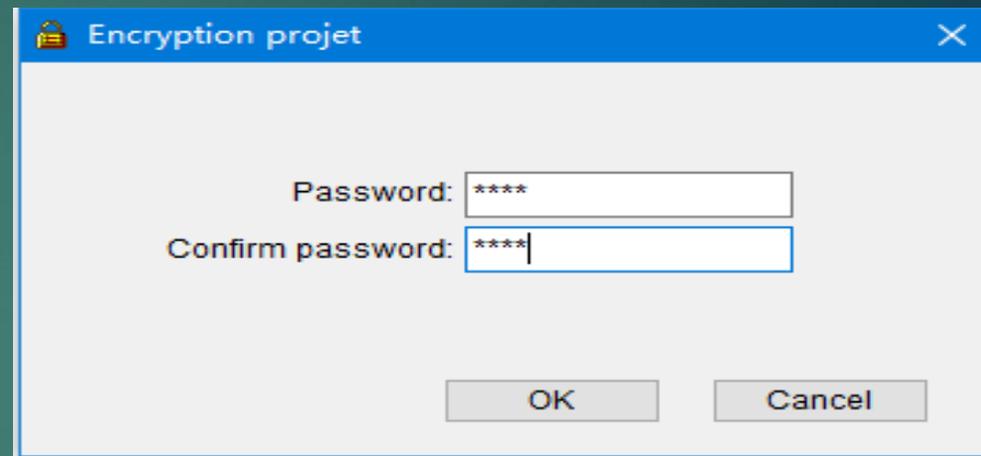
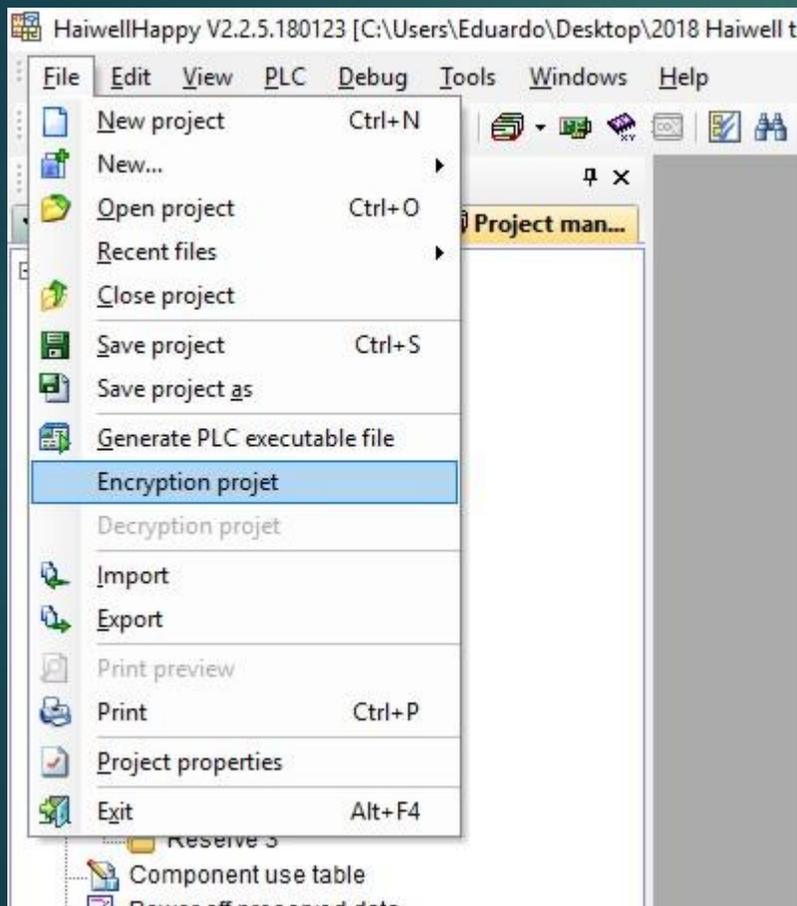
Confirm password: \*\*\*\*\*

OK Cancel

# Encriptação do projeto

8

- ▶ Clique em **Encryption project** e introduza uma senha.



- ▶ Observe que depois de introduzida a senha de encriptação, o programa “1 Motor control”, que está protegido por senha, desaparece do menu. O programa agora está escondido. Para o mesmo reaparecer é necessário desencriptar o projeto.

# Tabela de inicialização

8

Initial register table

Table name:

Start component:   Decimal  Hex  Float

Length:  (0 to 200)  Character

Power-off preservation: 1048 Point V1000 - V2047

Index	Component	Float (32 bit)	Hex	Component comments
1	V1200	12.34	0x414570A4	
2	V1201	0.0	0x00004145	
3	V1202	0.0	0x00000000	
4	V1203	0.0	0x00000000	
5	V1204	0.0	0x00000000	
6	V1205	0.0	0x00000000	
7	V1206	0.0	0x00000000	
8	V1207	0.0	0x00000000	
9	V1208	0.0	0x00000000	
10	V1209	0.0	0x00000000	
11	V1210	0.0	0x00000000	
12	V1211	0.0	0x00000000	
13	V1212	0.0	0x00000000	
14	V1213	0.0	0x00000000	
15	V1214	0.0	0x00000000	
16	V1215	0.0	0x00000000	
17	V1216	0.0	0x00000000	
18	V1217	0.0	0x00000000	
19	V1218	0.0	0x00000000	
20	V1219	0.0	0x00000000	
21	V1220	0.0	0x00000000	

Comments:

Password:

Confirm password:

Help Check code calculator OK Cancel

- ▶ Crie uma tabela de inicialização com o nome “Initial table test” com endereço inicial V1200 e tamanho de 30 elementos
- ▶ Introduza V1200 = 12.34
- ▶ Observe que o valor introduzido ocupa as posições V1200 e V1201

# Tabela de inicialização

8

Initial register table

Table name:  Start component:   Decimal  Hex  Float

Length:  (0 to 200)  Character

Power-off preservation: 1048 Point V1000 - V2047

Index	Component	16bits value	32bits value	Component comments
1	V1200	0x70A4	0x414570A4	
2	V1201	0x4145	0x1A2B4145	
3	V1202	<input type="text" value="1A2B"/>	0x00001A2B	
4	V1203	0x0000	0x00000000	
5	V1204	0x0000	0x00000000	
6	V1205	0x0000	0x00000000	
7	V1206	0x0000	0x00000000	
8	V1207	0x0000	0x00000000	
9	V1208	0x0000	0x00000000	
10	V1209	0x0000	0x00000000	
11	V1210	0x0000	0x00000000	
12	V1211	0x0000	0x00000000	
13	V1212	0x0000	0x00000000	
14	V1213	0x0000	0x00000000	
15	V1214	0x0000	0x00000000	
16	V1215	0x0000	0x00000000	
17	V1216	0x0000	0x00000000	
18	V1217	0x0000	0x00000000	
19	V1218	0x0000	0x00000000	
20	V1219	0x0000	0x00000000	
21	V1220	0x0000	0x00000000	

Comments:

Password:  Confirm password:

Buttons: Help, Check code calculator, OK, Cancel

- ▶ Selecione hexadecimal
- ▶ Introduza V1202 = 1A2B

# Tabela de inicialização

8

Initial register table

Table name:  Start component:   Decimal  Hex  Float

Length:  (0 to 200)  Character

Power-off preservation: 1048 Point V1000 - V2047

Index	Component	16bits value	32bits value	Component comments
1	V1200	28836	1095069860	
2	V1201	16709	439042373	
3	V1202	6699	2147424811	
4	V1203	32767	1584824319	
5	V1204	24182	31415926	
6	V1205	479	479	
7	V1206	0	0	
8	V1207	0	0	
9	V1208	0	0	
10	V1209	0	0	
11	V1210	0	0	
12	V1211	0	0	
13	V1212	0	0	
14	V1213	0	0	
15	V1214	0	0	
16	V1215	0	0	
17	V1216	0	0	
18	V1217	0	0	
19	V1218	0	0	
20	V1219	0	0	
21	V1220	0	0	

Comments:

Password:

Confirm password:

Buttons: Help, Check code calculator, OK, Cancel

- ▶ Selecione decimal
- ▶ Introduza V1203 = 32767
- ▶ Introduza V1204 = 31415926
- ▶ Observe que V1205 recebe parte do valor

# Tabela de inicialização

8

Initial register table

Table name:

Start component:   Decimal  Hex  Float

Length:  (0 to 200)  Character  Low-byte mode

Power-off preservation: 1048 Point V1000 - V2047

Index	Component	Character	Hex	Component comments
1	V1200	' '	0x70A4	
2	V1201	'EA'	0x4145	
3	V1202	'+'	0x1A2B	
4	V1203	'I'	0x7FFF	
5	V1204	'v'	0x5E76	
6	V1205	'?'	0x01DF	
7	V1206	AT	0x5441	
8	V1207	'^S'	0x535E	
9	V1208	'MS'	0x534D	
10	V1209	'=0'	0x303D	
11	V1210	'59'	0x3935	
12	V1211	'22'	0x3232	
13	V1212	'23'	0x3332	
14	V1213	'03'	0x3330	
15	V1214	'12'	0x3231	
16	V1215	' '	0x0000	
17	V1216	' '	0x0000	
18	V1217	' '	0x0000	
19	V1218	' '	0x0000	
20	V1219	' '	0x0000	
21	V1220	' '	0x0000	

Comments:

Password:

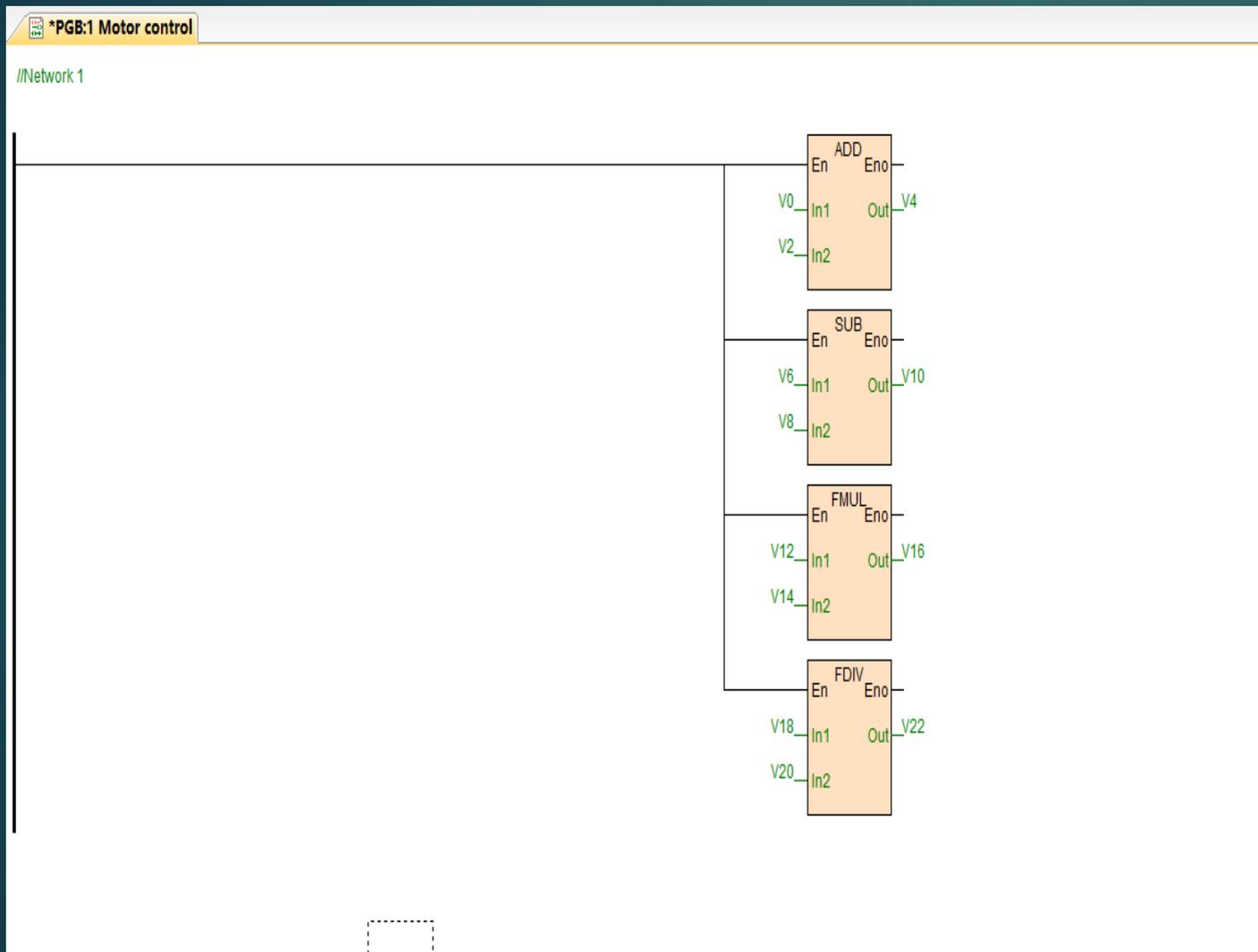
Confirm password:

Buttons: Help, Check code calculator, OK, Cancel

- ▶ Selecione Character
- ▶ Introduza V1206 = AT^SMS = 05922230312
- ▶ Com isso você aprendeu a criar e preencher uma tabela de inicialização com dados nos diversos formatos numéricos e na forma de caractere

# Instruções matemáticas

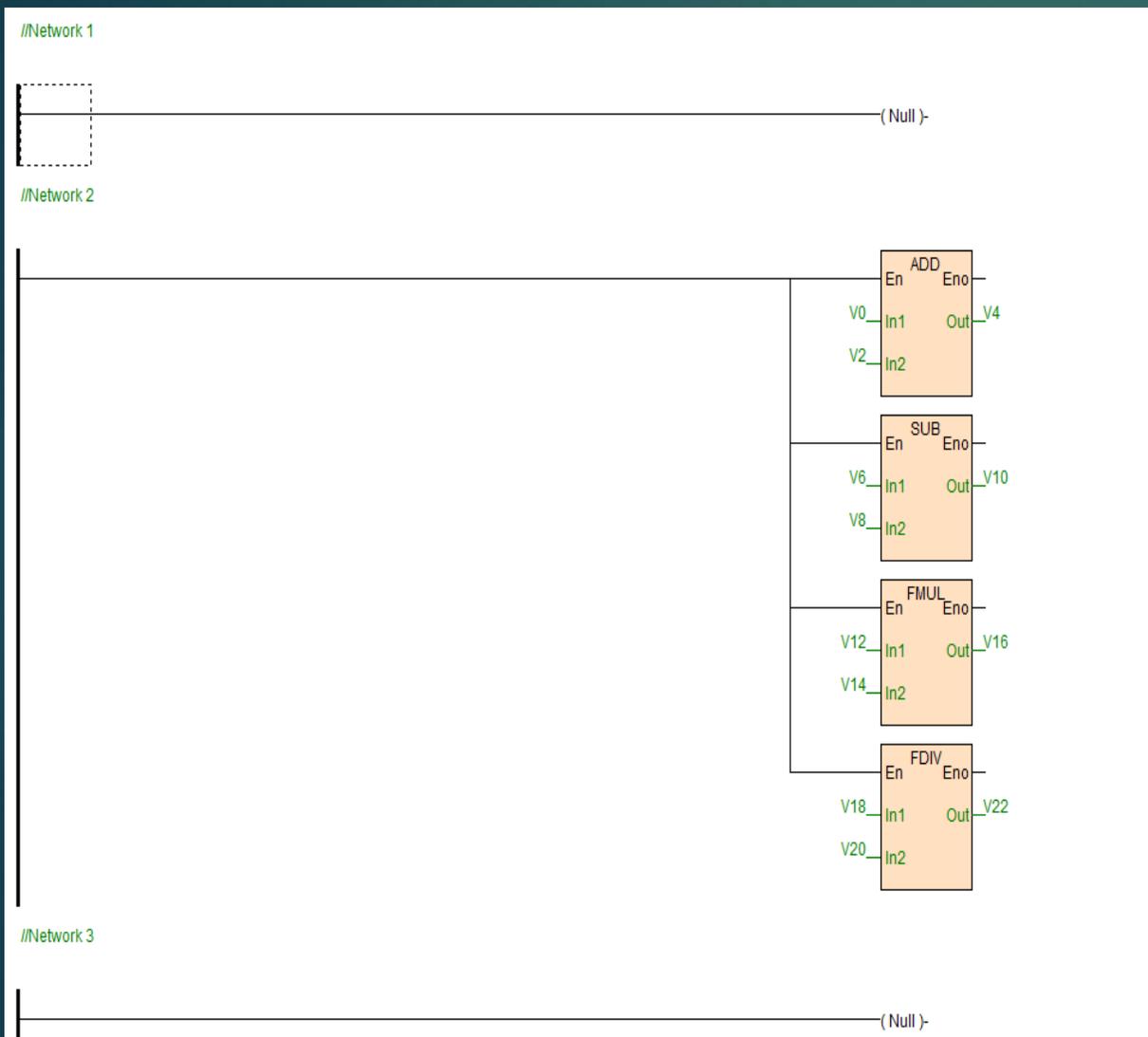
8



- ▶ Programe as seguintes operações matemáticas:
- ▶ ADD “ $V4 = V0 + V2$ ”
- ▶ SUB “ $V10 = V6 - V8$ ”
- ▶ FMUL “ $V16 = V12 * V14$ ”
- ▶ FDIV “ $V22 = V18 / V20$ ”

# Atalhos CTRL+I e CTRL+L

8



- ▶ Introduza uma linha antes da linha das operações matemáticas utilizando CTRL+I
- ▶ Introduza uma linha após a linha das operações matemáticas utilizando CTRL+L
- ▶ Uma lista com todas as teclas de atalho pode ser encontrada no HELP

Haiwell PLC programming software help

Ocultar Voltar Imprimir Opções

Sumário Índice Pesquisar

Digite a palavra-chave para localizar:

Programing operation manual

Listar Tópicos

Selecione o tópico para exibição:

- Hardware manual
- Product introduction
- Programming operation manual
- Quick start
- Remote module
- Simulate and online debugging

**Shortcut key**

HaiwellHappy programming software supply abundant shortcut key, during use programming operation easy and efficient ,shortcut key listing as follows:

Category	Shortcut key	Function
menu operation	Alt + F	Open "File" menu
	Alt + E	Open "Edit" menu
	Alt + V	Open "Search" menu
	Alt + P	Open "PLC" menu
	Alt + D	Open "Debug" menu
	Alt + T	Open "Tools" menu
	Alt + W	Open " window" menu
	Alt + H	Open "Help" menu
	Ctrl + N	New program project

# Comparadores e timer

8

## ▶ Introduza a linha abaixo

//Network 3 Testes com reles de comparacao e acionamento do timer



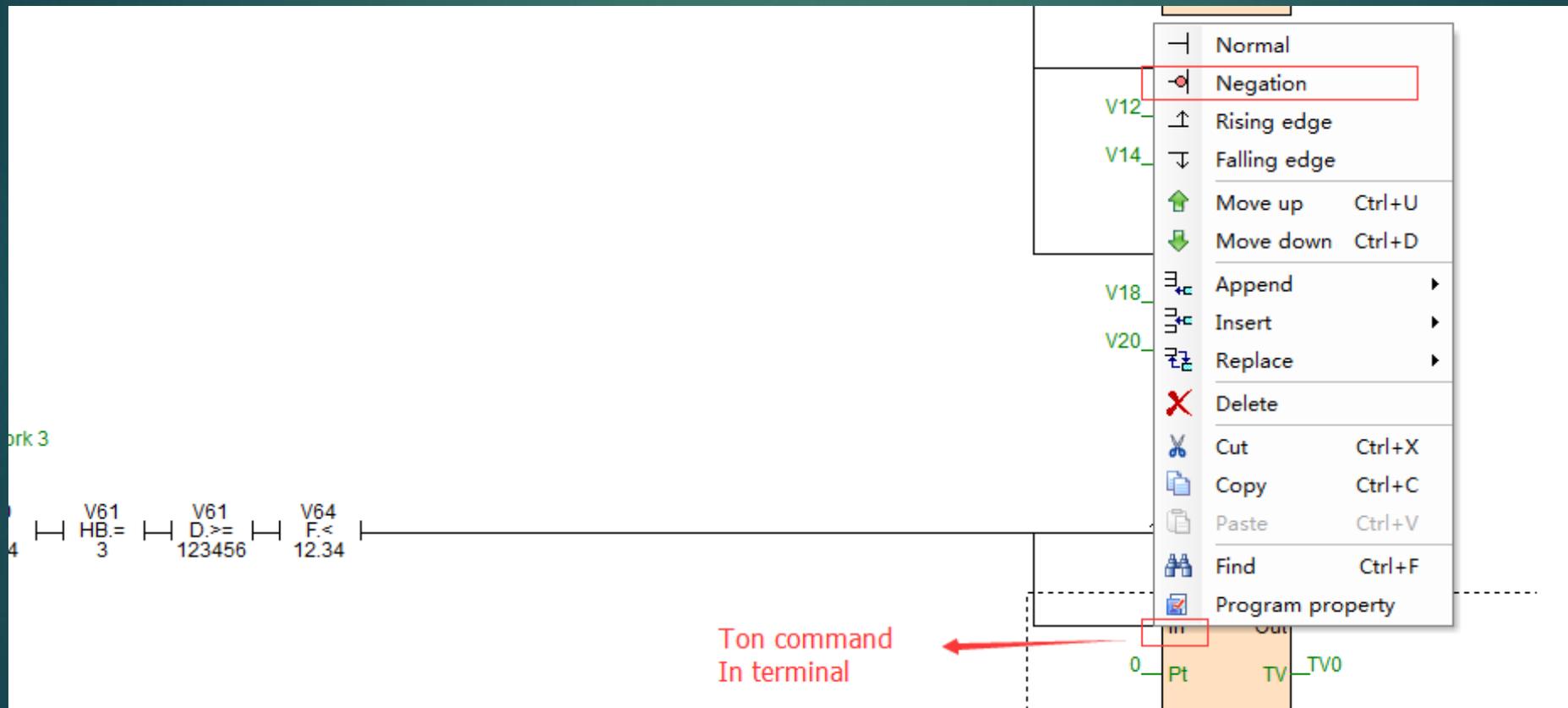
- ▶ Para tanto, na linha Network 3, clique em **serial connect switch** no menu superior ou pressione F9 para adicionar o relé. Clique com o botão direito para selecionar **16-bit comparison switch =**, digite V60 e 1234 respectivamente. Adicione um relé e selecione **High Bite compare equal switch =**, digite V61 e 3 respectivamente. Similarmente, adicione o relé **32-bit comparison switch > =**, e um relé **floating-point comparison switch <** e o valor 12,34, então introduza o comando **SET** associado a entrada Y0 com **rising edge**.



# Comparadores e timer

8

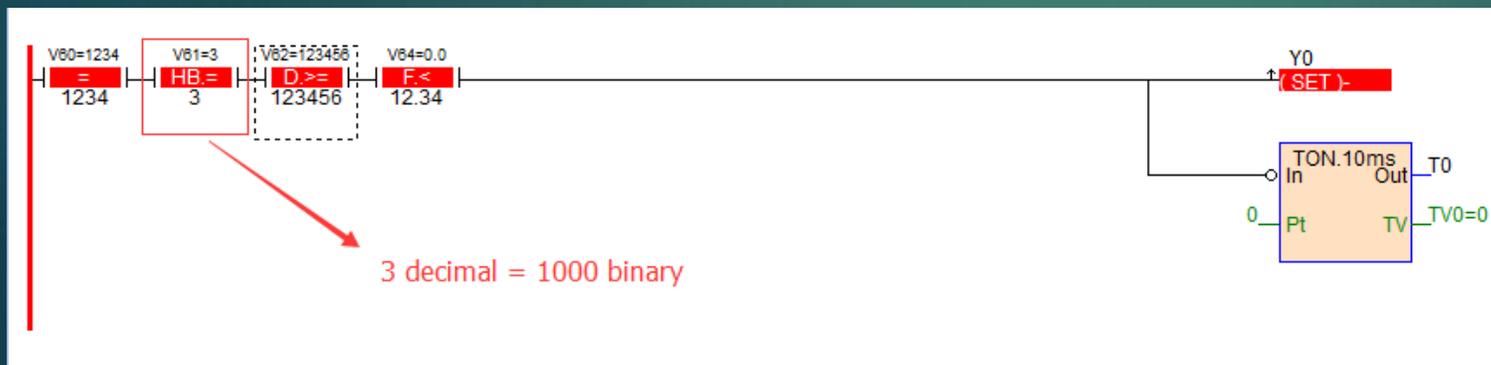
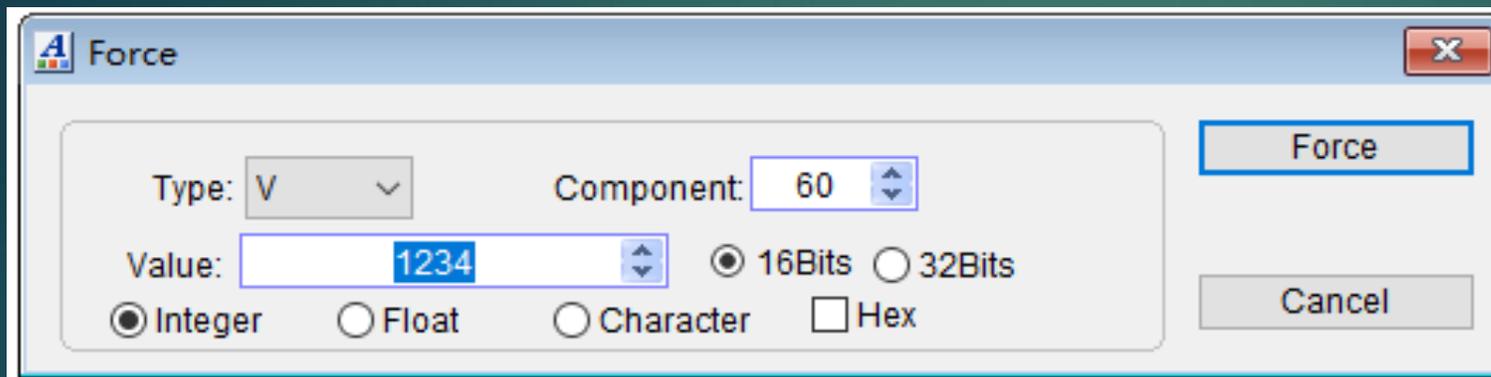
- ▶ Clique na conexão de entrada com o botão direito e selecione **Negation** para negar o sinal de acionamento do timer



# Simulação do funcionamento

8

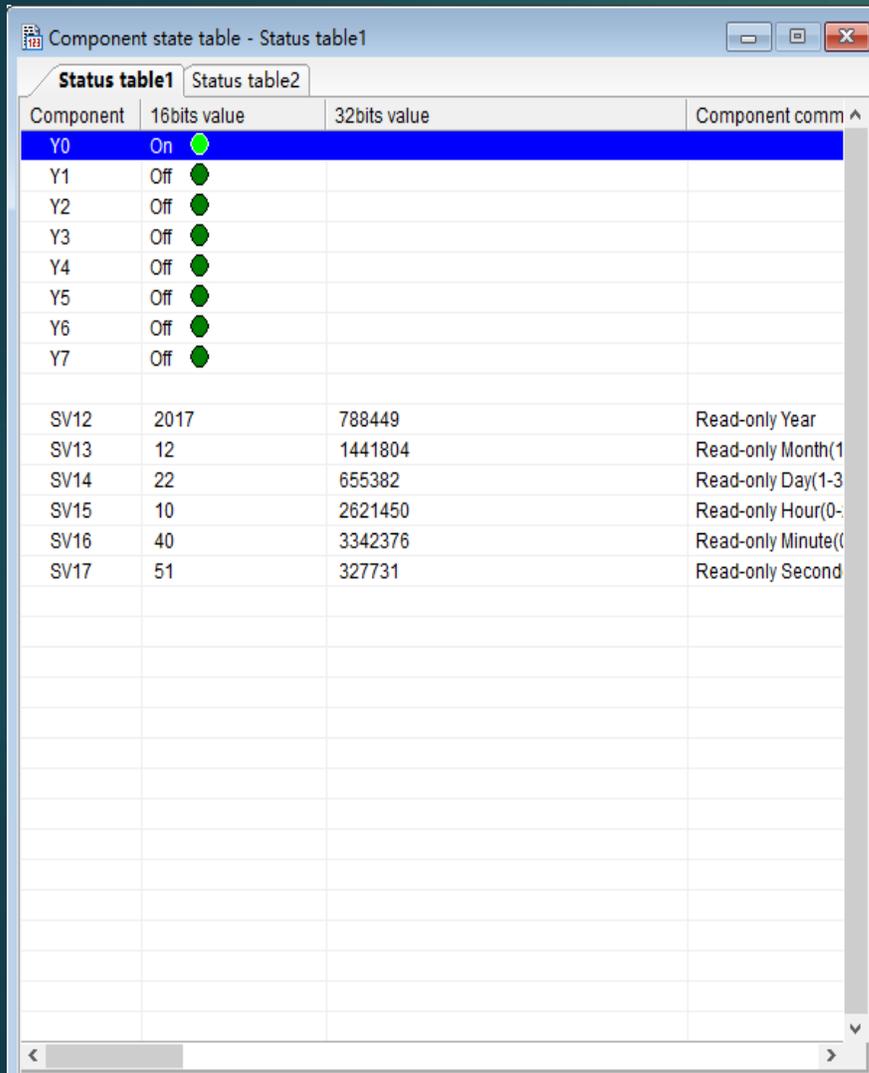
- ▶ Clique **Emulator start button** no menu superior e clique em V60 para forçar o valor 1234, similarmemente force os valores em V61, V62, V64



- ▶ Observe que Y0 irá ligar quando as condições forem atendidas

# Simulação do funcionamento

8



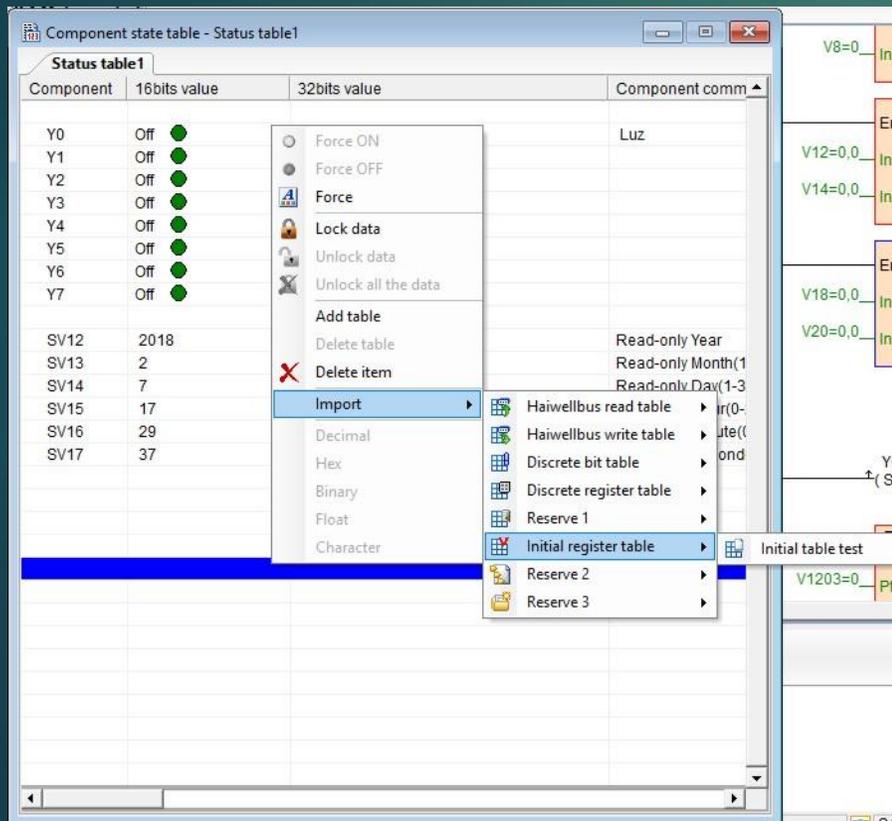
Component	16bits value	32bits value	Component comm ^
Y0	On 		
Y1	Off 		
Y2	Off 		
Y3	Off 		
Y4	Off 		
Y5	Off 		
Y6	Off 		
Y7	Off 		
SV12	2017	788449	Read-only Year
SV13	12	1441804	Read-only Month(1
SV14	22	655382	Read-only Day(1-3
SV15	10	2621450	Read-only Hour(0-
SV16	40	3342376	Read-only Minute(t
SV17	51	327731	Read-only Second

- ▶ Ainda em simulação, abra a janela **Status table 1** e introduza os valores Y0 a Y7 (digitando Y0-7 as 8 primeiras saídas são criadas na tabela)
- ▶ Adicione os variáveis de sistema do relógio de tempo real SV12 a SV17 (digitando SV12-17 as 6 variáveis são criadas na tabela)

# Importação de tabela de monitoração

8

- ▶ Importe a tabela de inicialização para a tabela de monitoração
- ▶ Isso irá criar uma nova aba **Status Table 2** com os valores definidos na tabela que havíamos criado **Initial table test**



The screenshot shows the 'Component state table - Status table2' window. The table contains the following data:

Component	16bits value	32bits value	Component comm
V1200	28836	1095069860	
V1201	16709	439042373	
V1202	6699	2147424811	
V1203	32767	1584824319	
V1204	24182	31415926	
V1205	479	1413546463	
V1206	21569	1398690881	
V1207	21342	1397576542	
V1208	21325	809325389	
V1209	12349	959787069	
V1210	14645	842152245	
V1211	12850	858927666	
V1212	13106	858796850	
V1213	13104	842085168	
V1214	12849	12849	
V1215	0	0	
V1216	0	0	
V1217	0	0	
V1218	0	0	
V1219	0	0	
V1220	0	0	
V1221	0	0	
V1222	0	0	
V1223	0	0	
V1224	0	0	
V1225	0	0	
V1226	0	0	
V1227	0	0	
V1228	0	0	
V1229	0	0	

# Rotina de interrupção por tempo

8

- ▶ Pare o simulador, delete a linha 1
- ▶ Introduza uma nova linha com CTRL+L e configure um timer T252 com base de tempo de 1 ms e preset de 125
- ▶ Adicione a instrução ATCH para chamada de sub-rotina acionada pelo timer T252



# Rotina de interrupção por tempo

8

- ▶ Clique no organizador em **Int program** com o botão direito e clique em **New int program**
- ▶ Configure o bloco como abaixo
- ▶ Está criado o bloco da rotina por interrupção

Int program Project properties

Block name:

Language:

Password:

Confirm password:

Comments:

Type:

- Main program
- Sub program
- Int program

Language:

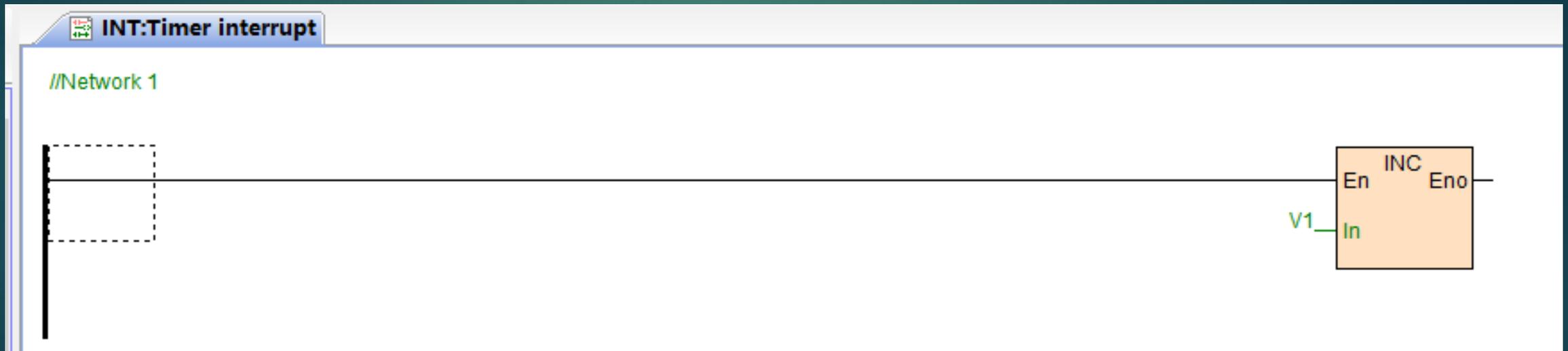
- LD
- FBD
- IL
- SFC
- ST

OK Cancel

# Rotina de interrupção por tempo

8

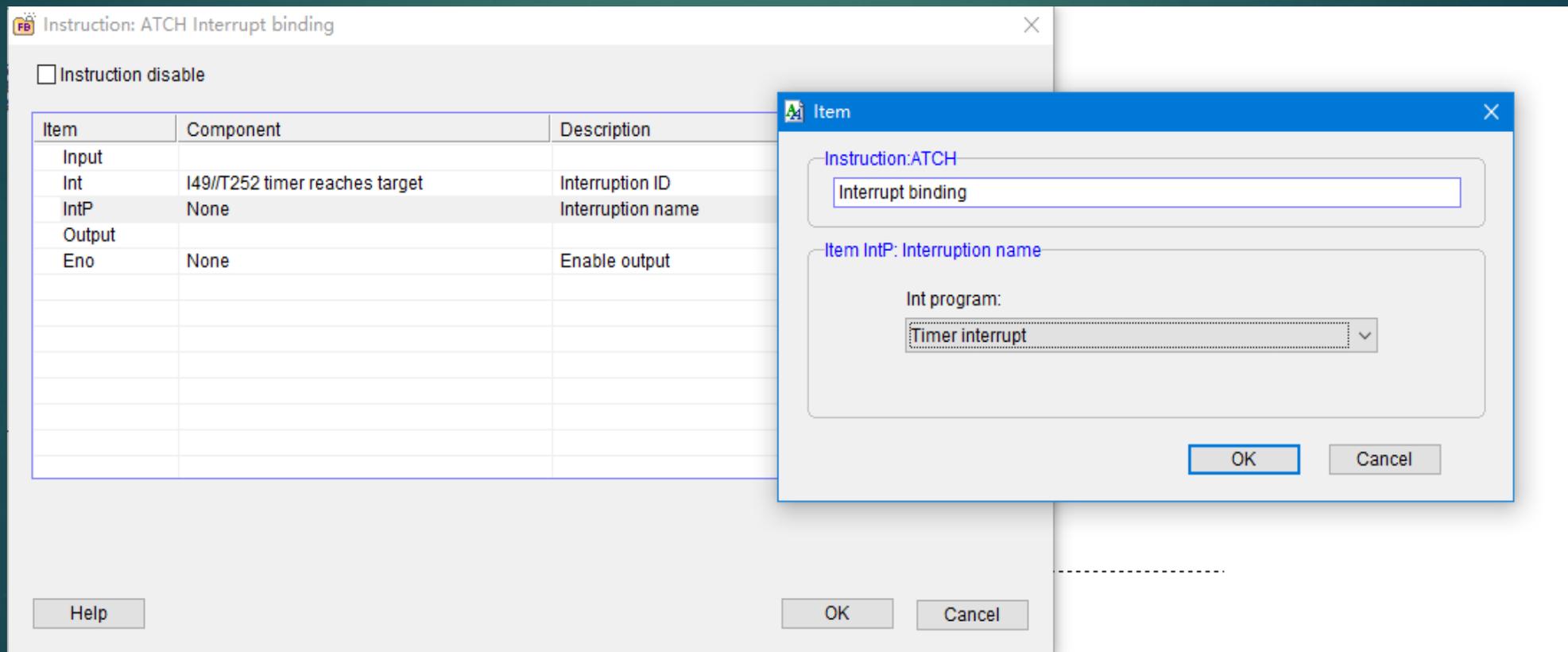
- ▶ Programe a rotina de interrupção de forma a incrementar V1 a cada vez que for acionada



# Rotina de interrupção por tempo

8

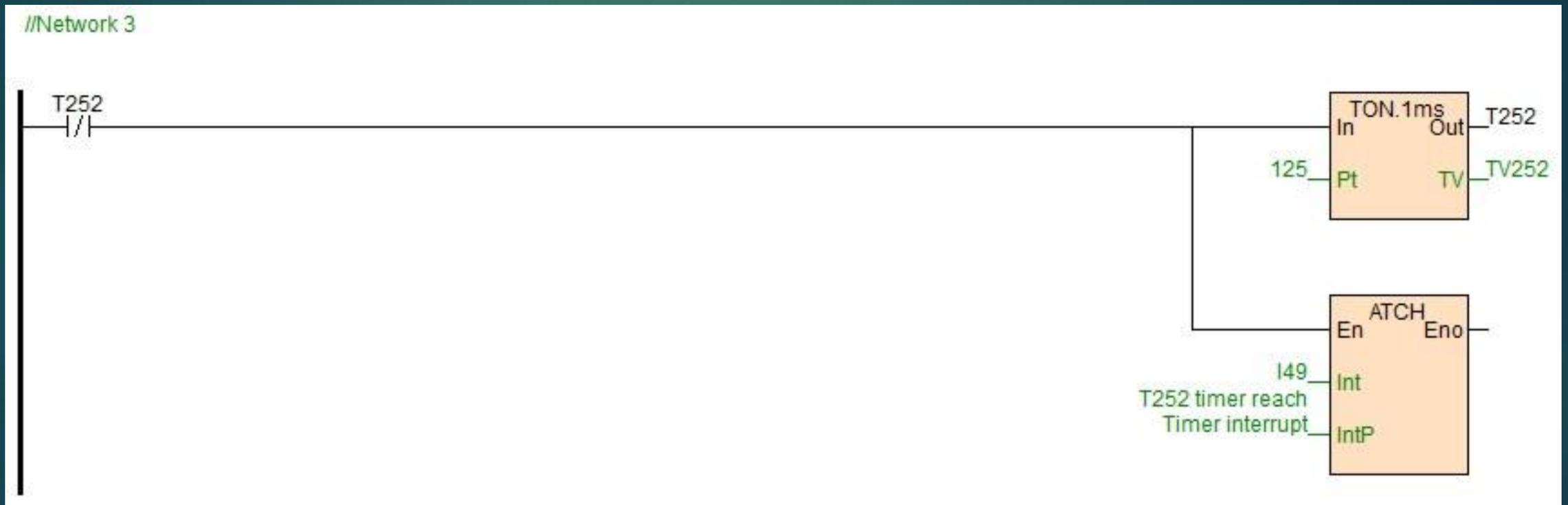
- ▶ No programa principal **1 Motor control**, clique duplo na instrução ATCH, selecione a I49



# Rotina de interrupção por tempo

8

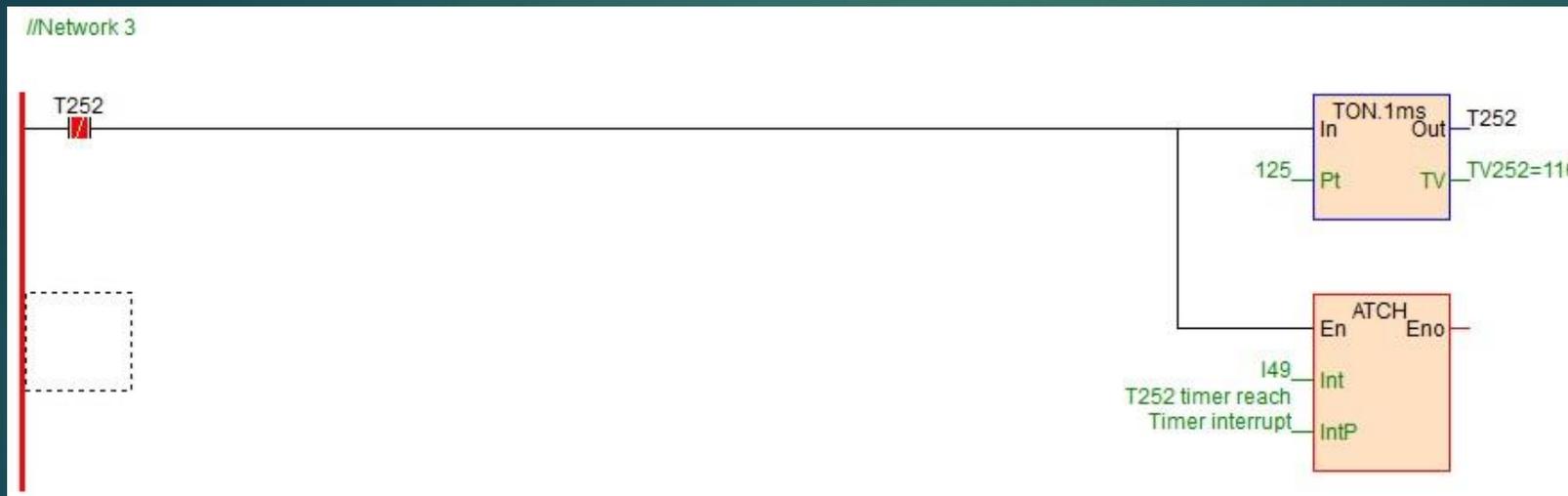
- ▶ Introduza o relé normalmente fechado T252 no início da linha para que o timer 252 seja resetado a cada 125 ms



# Rotina de interrupção por tempo

8

- ▶ Entre no simulador e observe que o timer é reiniciado a cada 125 ms quando o relé T252 pulsa
- ▶ A cada reinicialização a rotina de interrupção é acionada



- ▶ Observe que o valor de V1 na rotina de interrupção é incrementado a cada vez que a rotina é acionada



# Sub-rotina

8

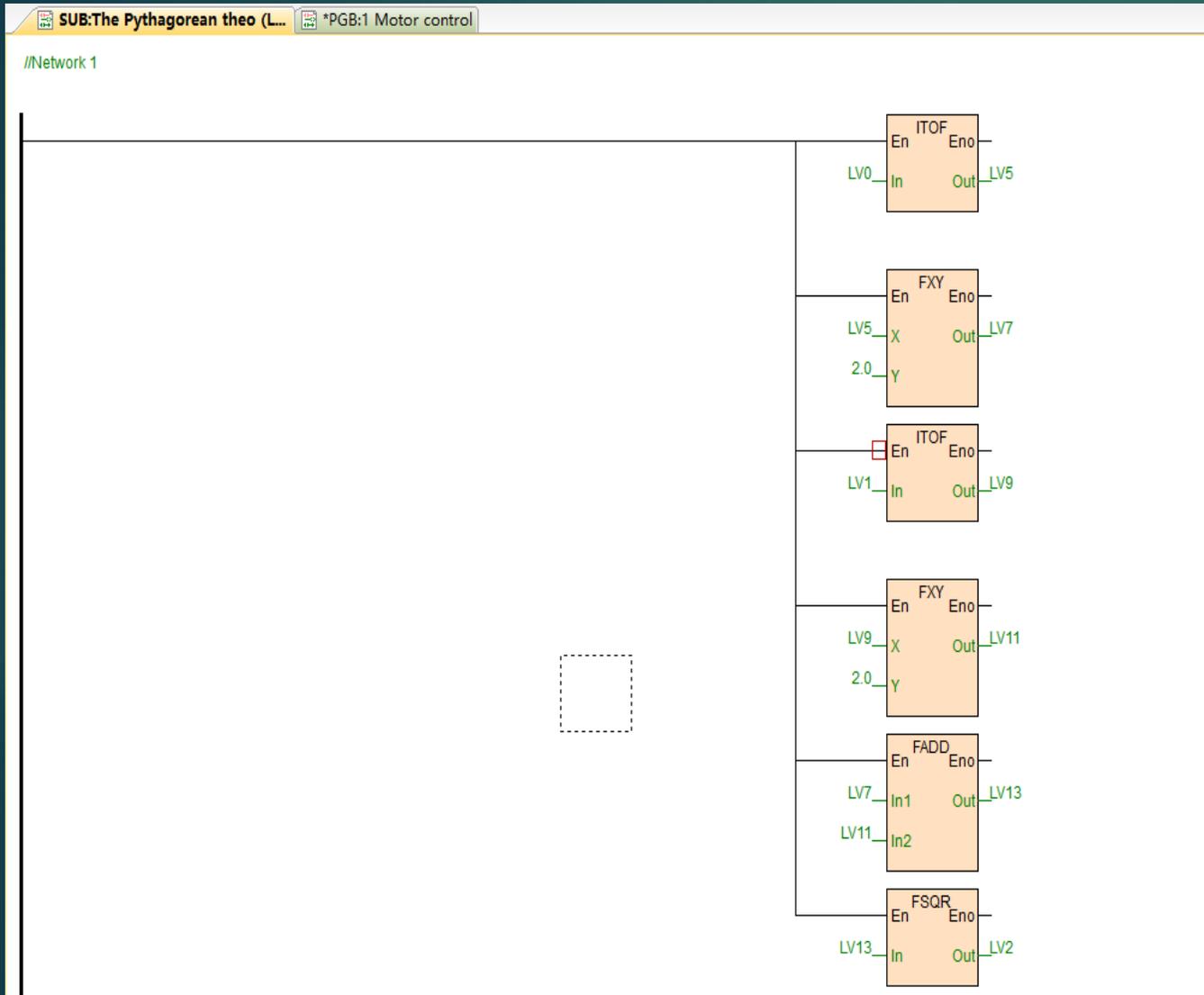
- ▶ Vamos agora criar uma sub-rotina para calcular a hipotenusa de um triângulo retângulo, dados dois catetos (  $a^2 = b^2 + c^2$  ), Teorema de Pitagoras

Component	Par. name	Par. type	Data type	Comments
LV0	LV0	IN	INT	
LV1	LV1	IN	INT	
LV2	OUT	OUT	REAL	

- ▶ Clique em **subroutine** no menu lateral, abra a janela de configuração e dê o nome "The Pythagorean Theo", clique **Append** e adicione LV0 e LV1 (INT) e LV2 (REAL) como ao lado

# Sub-rotina

8

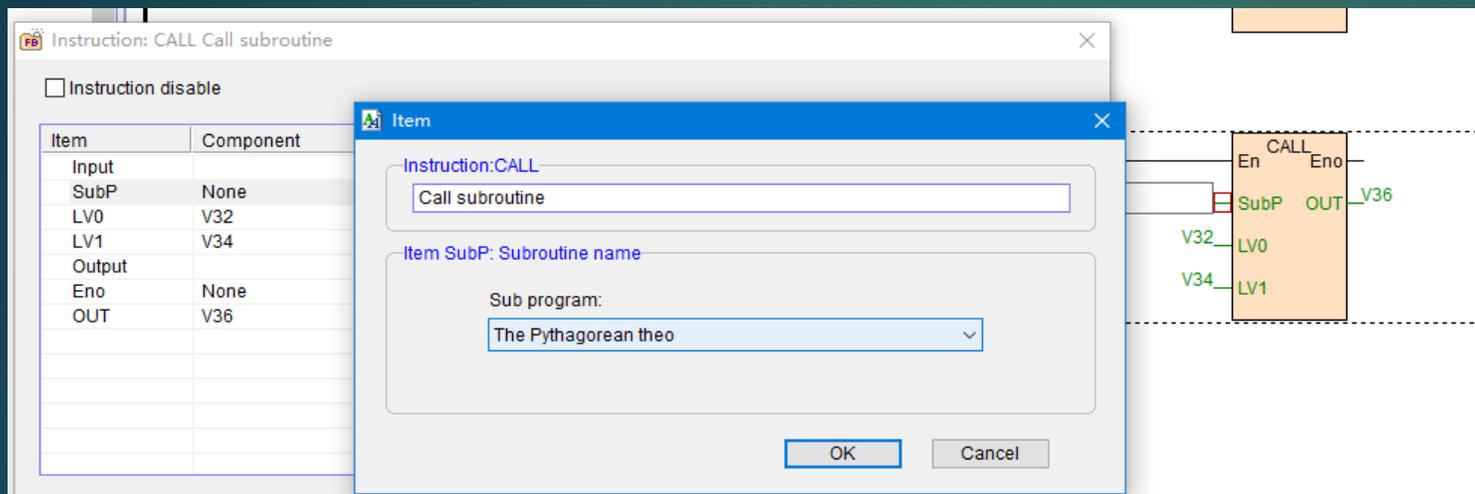


- ▶ Introduza as instruções ITOF, FXY, FADD e FSQR na sub-rotina como ao lado

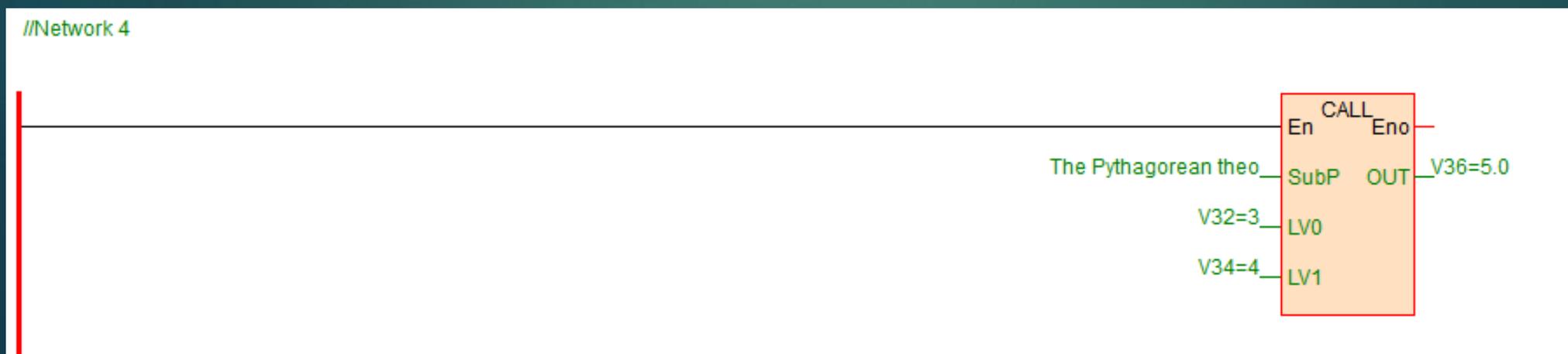
# Sub-rotina

8

- ▶ No programa **1 Motor Control** crie a chamada da sub-rotina com a instrução CALL



- ▶ Introduza as variáveis V32 e V34 como entradas e V36 como saídas
- ▶ Entre em modo simulação e atribua valores a V32 e V34
- ▶ Visualize a hipotenusa calculada em V36



# Monitor gráfico de tendência

8

- ▶ Monitore a variável interna SV17 (segundos) do relógio de tempo real e observe a evolução do valor
- ▶ Inicie o simulador e clique em **Trend monitor** e digite SV17 no primeiro campo
- ▶ Altere o **Upper limit** para 60 e observe a evolução gráfica do contador de segundos

The screenshot shows the 'Trend monitor' window in a simulation environment. The window title is 'Trend monitor'. Below the title bar, there are several tabs: 'Message window', 'Trend monitor', 'The table of lock data (0)', and 'Hardware simulation windows \_ Simulation status 2017/12/25 10:29:30'. The 'Trend monitor' tab is active.

On the left side, there is a table with the following columns: Component, Value, Lower limit, Upper limit, 32Bits, and Float. The first row is checked and shows 'SV17' with a value of 30, a lower limit of 0, and an upper limit of 60. The other rows are empty.

On the right side, there is a graph area with a green grid. The graph shows a sawtooth pattern, indicating a counter that increases linearly and then resets. The x-axis represents time, and the y-axis represents the value of SV17. The graph is currently set to a 5-minute scale.

At the bottom of the window, there is a status bar with the following information: 'Ready', 'Simulation', 'PLC Offline', 'Currently PLC:None', 'Scan time:1.0 ms', and 'Networks: 1 of 3'.

# Aula 8 – Assuntos apresentados

8

O conteúdo desta aula teve como objetivo preparar os alunos para a avaliação on-line para a obtenção do certificado de participação no treinamento CURSO DE AUTOMAÇÃO UTILIZANDO O CLP HAIWELL.

Para acompanhar esta aula é necessário ter concluído todas as anteriores.

Siga passo-a-passo a aula 8 para estar preparado para a avaliação.

Veja ao lado os assuntos desta aula.

- ▶ Iniciar um novo projeto
- ▶ Definição da área de memória retentiva
- ▶ Criação de blocos de programa
- ▶ Encriptação de projetos
- ▶ Criação de tabelas de inicialização de dados
- ▶ Adição de blocos matemáticos de função
- ▶ Instruções básicas de temporização
- ▶ Simulação do programa
- ▶ Monitoração do relógio de tempo real

OBRIGADO POR ACOMPANHAR NOSSO CURSO!

VISITE NOSSO SITE E FAÇA O DOWNLOAD DOS MANUAIS E SOFTWARES.

A VERSÃO PDF DESTA AULA PODE SER ENCONTRADA NO SEGUINTE LINK:

[HTTP://WWW.ALFACOMP.IND.BR/PROGRAMACAO-HAIWELL-PROD-81.HTML](http://www.alfacomp.ind.br/PROGRAMACAO-HAIWELL-PROD-81.HTML)

ATÉ A PRÓXIMA AULA