بسم الله الرحمن الرحيم

هذه مقدمة أوامر لغة البرمجة SIMATIC STEP 7
سائلا المولى عز وجل أن ينفع بها المختصين في شتى المجالات ولا تنسونا من صالح الدعاء

مهندس صالح سعيد بوحليقة
محطة كهرباء الزويتينة الغازية - ليبيا
Email- zwuitina@yahoo.com

أوامر لغة البرمجة SIMATIC STEP 7

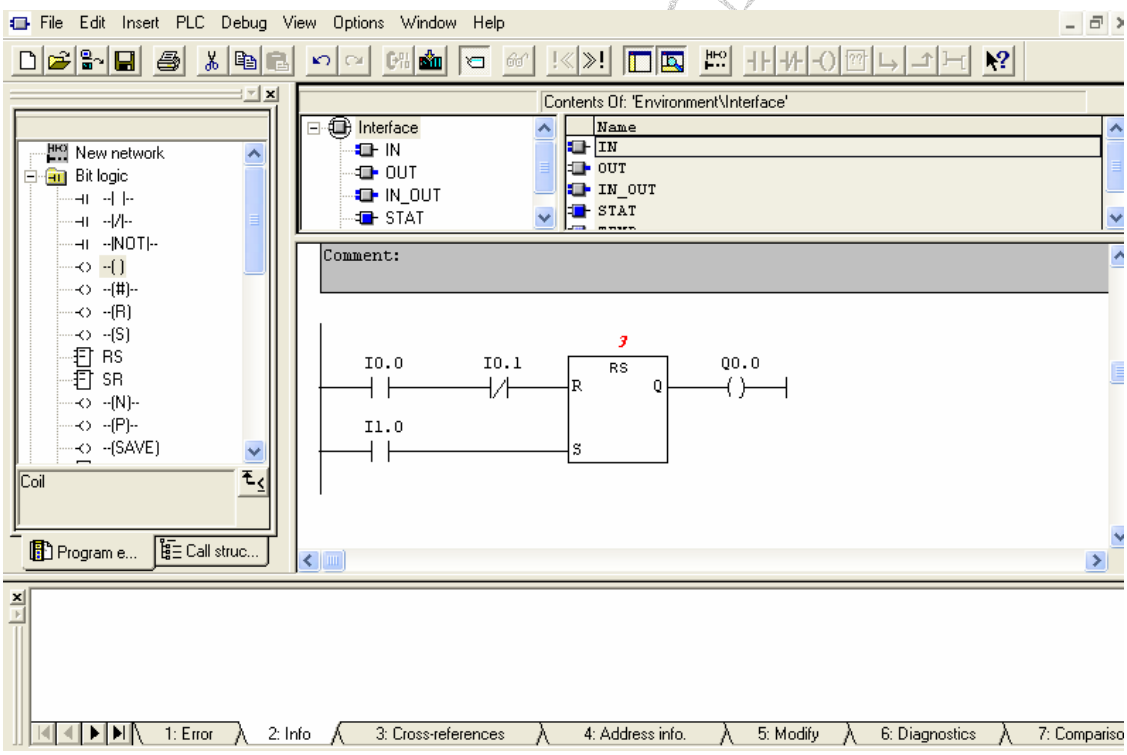
إعداد مهندس صالح سعيد بوحليقة

لغة البرمجة SIMATIC STEP 7

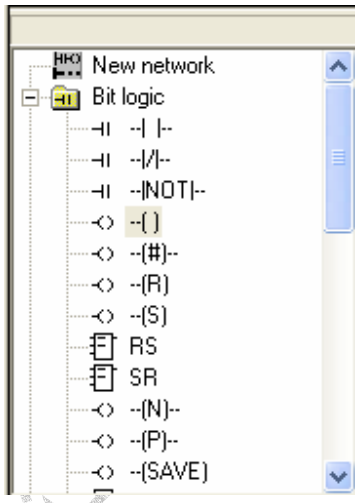
تعتبر منظومة التحكم step7 من أكثر منظومات التحكم شيوعا في العالم ويرجع الفضل في ذلك إلى تميزها بسهولة التركيب والبرمجة والاستخدام حيث كانت البداية مع ظهور منظومة التحكم step 5 التي تمتاز بسرعة العالية في التطبيقات ومن ثم جاءت منظومة التحكم step7 كمطور للمنظومة ثم تم تطوير المنظومة إلى إصدارتها المتلاحقة إلى أن وصلة إلى step7 400 وفيما يلي مراحل تطوير المنظومة

STEP 5
STEP 7 200
STEP 7 300
STEP 7 400

الشكل العام للغة البرمجة



أوامر Bit logic

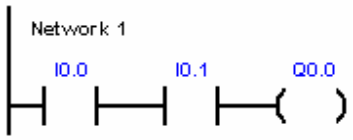


Output

الأمر Output وهو خرج الإشارة ويرمز له بالرمز Q

Standard Contact

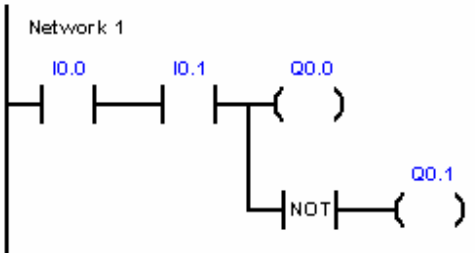
الأمر Standard Contact وهو يمثل نقطة تلامس مفتوحة إلا عندما تكون قيمة العنوان المخصص لها 1



في المثال اعلاه لاتكون قيمة الخرج Q0.0 1 إلا إذا كانت قيمة الدخل I0.0,I0.1 كلاهما يساوى 1 حيث يمكن عن طريقها الحصول على بوابة AND,OR الخ كما في المثال أعلاه

NOT

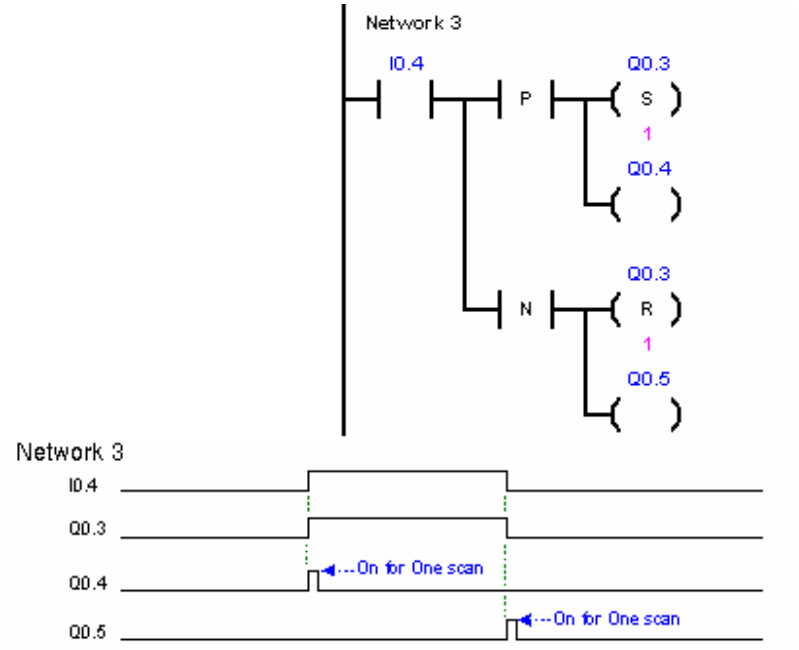
الأمر NOT وهو يمثل عكس الإشارة حيث إذا كانت الإشارة 1 يتم عكسها إلى 0



في المثال أعلاه تكون قيمة الخرج Q0.1 عكس قيمة الخرج Q0.0

Positive, Negative Transition

الأمر Positive, Negative Transition وهو عبارة عن عكس أقطاب الإشارة بوضع + / -

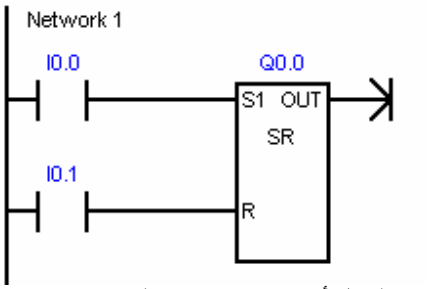


يمكن فهم الأمر بمخطط الإشارات أعلاه حيث يتم عكس قطب الإشارة للبوابة S R عندما تكون إشارة الدخل I0.4 1 تكون إشارة الخرج Q0.3 1 عندها يتم عمل نبضة من الخرج Q0.4 وعندما تكون إشارة الخرج Q0.3 0 عندها يتم عمل نبضة من الخرج Q0.5

البوابة S R

البوابة S R وتسمى النطاط ويمكن التعريف بالبوابة كلاتى

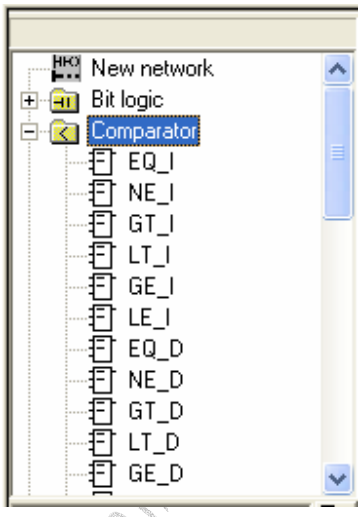
S1	R	Out (Bit)
0	0	Previous state
0	1	0
1	0	1
1	1	1



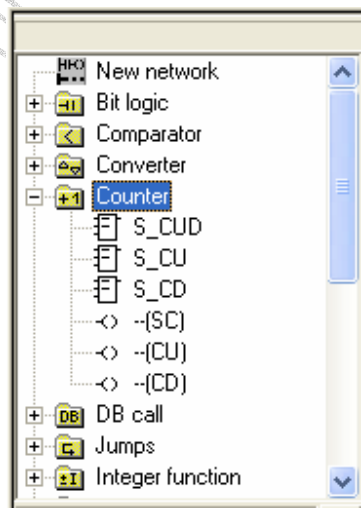
في المثال أعلاه يكون خرج البوابة 1 عندما يكون قيمة كلا من الدخل I0.0 تساوى 1 والدخل I0.1 تساوى 0 أو كلا منهما تساوى 1

أوامر المقارنة

تستخدم أوامر المقارنة في لغة البرمجة STEP 7 مثل جميع لغات البرمجة الأخرى واهم هذه الأوامر هي اكبر من اويساوى ، اكبر من ,اصغر من أو يساوى ، اصفر من , يساوى

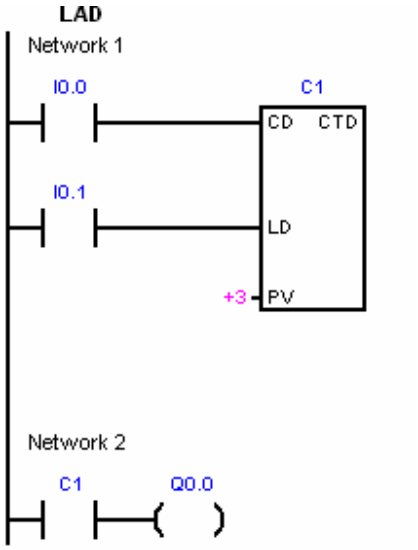


أوامر العد

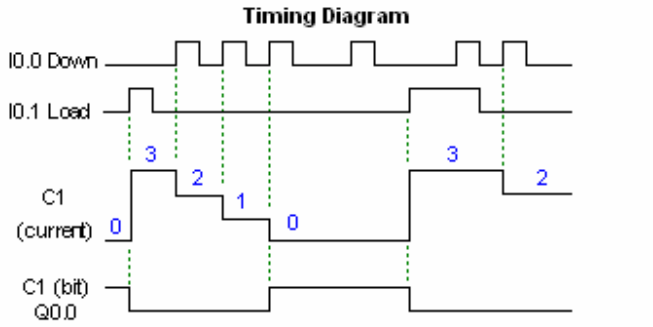


تستخدم العدادات لتخزين قيم أو عد عمليات معينه وتوجد انواع من العدادات منها عداد تصاعدي وعداد تنازلي الخ
وفى المثال أدناه يستخدم عداد تنازليCTD يقوم بالعد التنازلي كلما كانت قيمة الدخل I0.0 تساوى 1 كلا على حدي ماداما قيمة الدخل I0.1 تساوى 0 حيث تعتبر قيمة الدخل I0.1 هي إعادة تصفير العداد

CD إشارة بدء العد
LD إشارة إلغاء العد
PV قيمة العداد



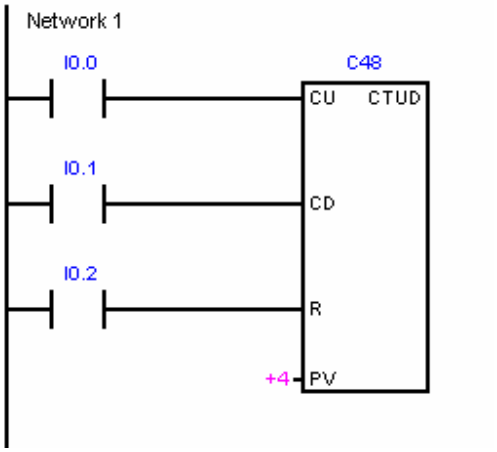
ويمكن فهم المثال أعلاه بالمخطط الإشارات الأتي



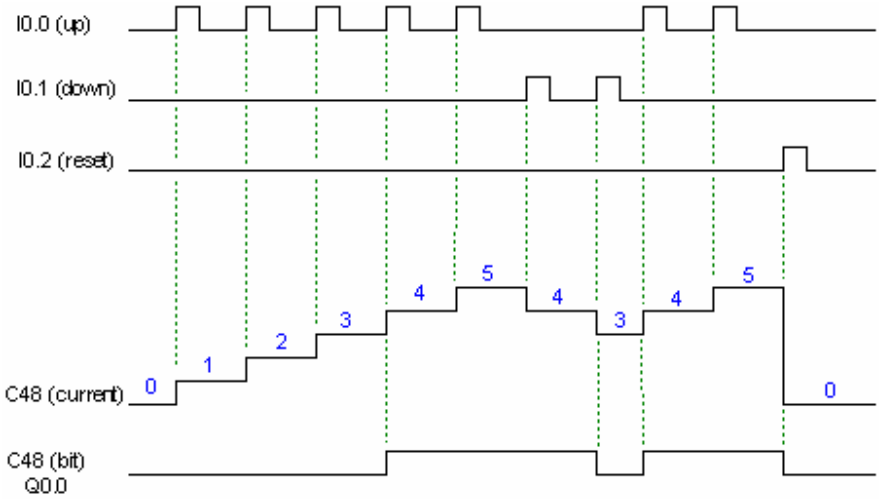
حيث يتم العد التنازلي كلما تغيره قيمه الدخل I0.0 مع كون قيمه الدخل I0.1 تساوى 0 وعند نهاية العد يكون قيمة خرج العداد 1 حيث يعتبر الدخل I0.1 هو المتحكم في تصفير العداد مهما تغيره قيمة الدخل I0.0

عداد تصاعدي تنازلي

وفيه يتم العد التصاعدى والتنازلي في إن واحد على حسب إشارات الدخل

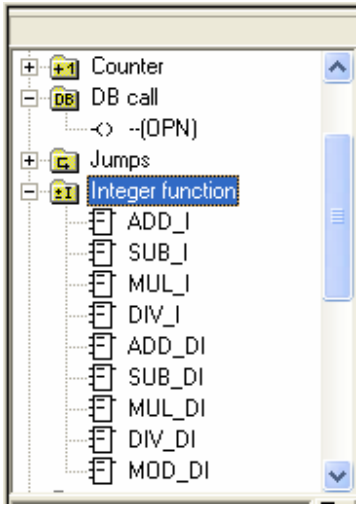


في المثال أعلاه يتم استخدام عداد تنازلي تصاعدي مع ثلاث إشارات دخل وقيمة عد أربع مراحل

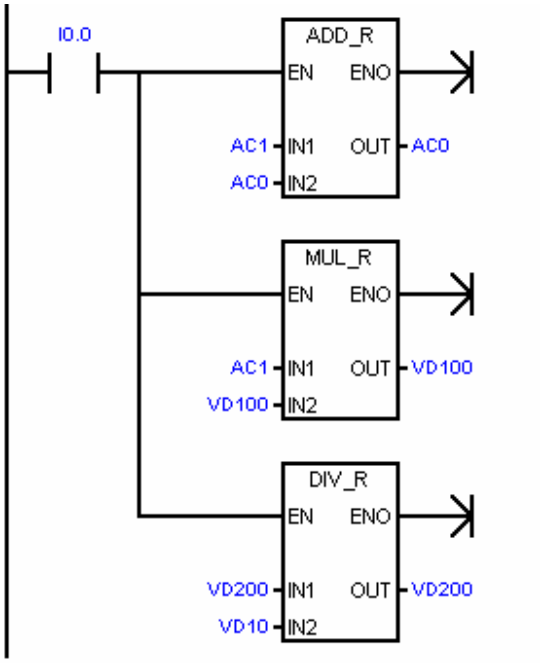


ويمكن فهم المثال أعلاه بالمخطط الإشارات أعلاه حيث يتم العد التصاعدي كلما تغيرت قيمة الدخل I0.0 من 0 إلى 1 يتم العد التنازلي كلما تغيرت قيمة الدخل I0.1 من 0 إلى 1 ويكون خرج العداد 1 عند وصول العداد إلى أقصى مرحل العد المخصصة ويتم تصفير العداد عن طريق الدخل I0.2

أوامر العمليات الحسابية



وهى مثل غيرها في جميع لغات البرمجة ويرمز لها ب ADD للجمع و MUL للضرب و SUP للطرح و DIV للقسمة



في المثال أعلاه يتم جمع قيمة المتغير AC0,AC1 وتخزين ناتج الجمع في المتغير AC0 و ضرب قيمة المتغير VD100,AC1 وتخزين ناتج الجمع في المتغير VD100 وقسمة قيمة المتغير VD200,VD10 وتخزين ناتج الجمع في المتغير VD200

وهنا يمكن توضيح نتائج البرنامج أعلاه

	IN1		IN2	OUT
Add Data	4000.0	+	6000.0	10000.0
Data Address	AC1		AC0	AC0
Multiply Data	400.0	*	200.0	80000.0
Data Address	AC1		VW102	VW100
Divide Data	4000.0	/	41.0	97.5609
Data Address	VW200		VW10	VW200

كما تحتوى لغة البرمجة STEP7على بعض العمليات الحسابية الخاصة مثل √,ROOT ,LN ,TAN,COS,SIN

أوامر العمليات المنطقية

Invert byte

وفية يتم عكس قيمة الدخل (byte) إلى عكس قيمته الأصلية مثال 1111 0110 إلى 1001 0000

Invert word

وفية يتم عكس قيمة الدخل (word) إلى عكس قيمته الأصلية مثال 1111 0110 0000 1001 إلى 0110 1111 1001 0000

Invert Double word

وفية يتم عكس قيمة الدخل (double word) 32 bit إلى عكس قيمته الأصلية

الأمر AND

وفيه لا تكون قيمة الخرج 1 إلا إذا كانت جميع قيم الدخل تساوى 1

الأمر OR

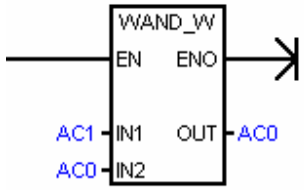
وفيه تكون قيمة الخرج 1 إذا كانت إحدى قيم الدخل تساوى 1

الأمر Exclusive OR

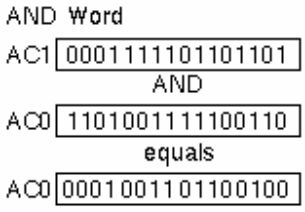
وفيه لايكون الخرج 1 إلا إذا كانت قيم الدخل غير متساوية اى واحدة 1 وأخرى 0

AND Word

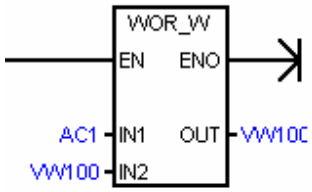
وفيه يتم تطبيق الأمر AND على Word Bit16 كما في المثال أدناه



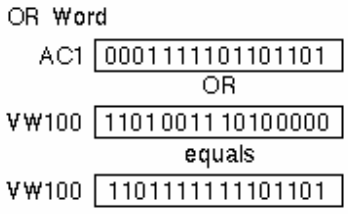
كما في المثال أعلاه يتم تطبيق الأمر AND على Word AC1,AC0 وتخزين الناتج في الخرج AC0 وتكون النتيجه كلاتى



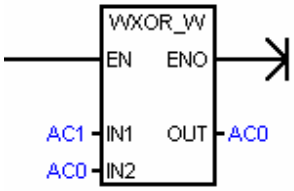
وفيه يتم تطبيق الأمر OR على Word Bit16 كما في المثال أدناه



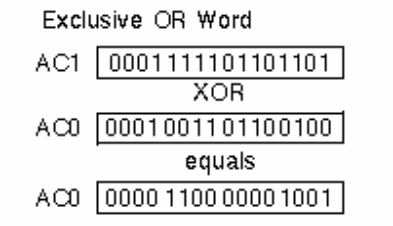
كما في المثال أعلاه يتم تطبيق الأمر OR على Word AC1,VW100 وتخزين الناتج فى الخرج VW100 وتكون النتيجة كلاتى



وفيه يتم تطبيق الأمر Exclusive OR على Word Bit16 كما في المثال أدناه

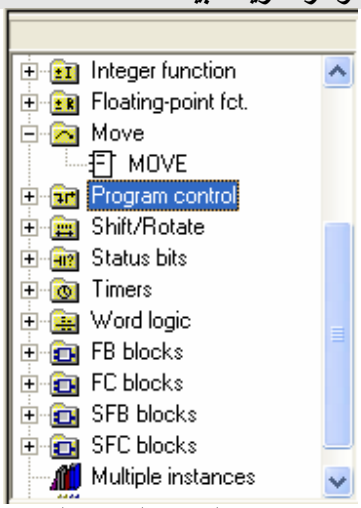


كما في المثال أعلاه يتم تطبيق الأمر Exclusive OR على Word AC1,AC0 وتخزين الناتج في الخرج AC0 وتكون النتيجة كلاتى

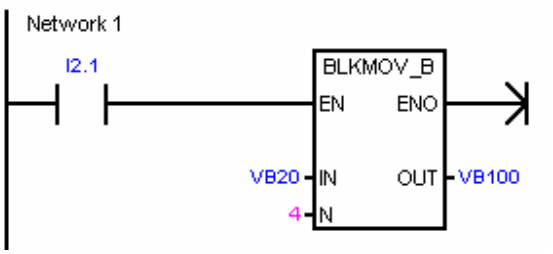


أوامر AND Double Word, OR Double Word, Exclusive OR Double Word
وهى مثل سابقتها إلا أنها تستخدم Bit 23 Double Word بدلا من Word
أوامر AND Byte, OR Byte, Exclusive OR Byte
وهى مثل سابقتها إلا أنها تستخدم Bit 8 Byte بدلا من Word

أوامر تحريك البيانات



وفيها يتم تحريك قيم البيانات إلى قيم اعلي أو اقل



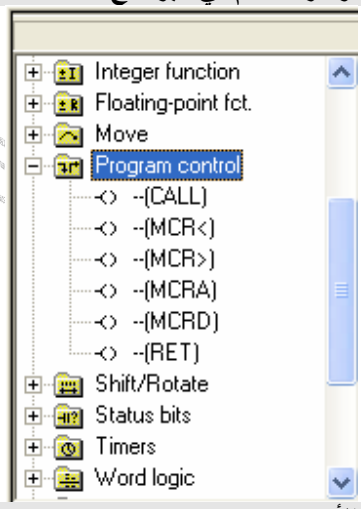
في المثال أعلاه يتم تحريك قيمة إشارات الدخل من VB20 إلى VB23 بزيادة مرة واحدة لكل إشارة
ملاحظة تم زيادة 4 إشارات دخل نظرا لاختيار N = 4

وكذالك الحال لإشارات الخرج

Array 1 Data	30	31	32	33
Data Addresses	VB20	VB21	VB22	VB23
Block Move Execution				
Loads Array 2				
Array 2 Data	30	31	32	33
Data Addresses	VB100	VB101	VB102	VB103

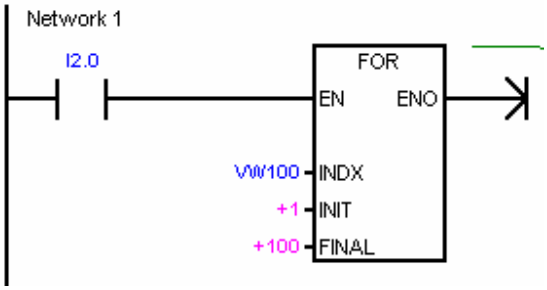
ويمكن تطبيق الأمر ليشمل تحريك إل byte, word and Double Word

أوامر التحكم في البرنامج



الأمر FOR

وهو يستخدم لعمل دائرة مستمرة للبرنامج إلى إن يتم تطابق الشرط لإنهاء الدائرة



في المثال أعلاه يتم زيادة 1 صحيح إلى قيمة الدخل VW100 وتتكرر العملية إلى إن تتم الزيادة 100 مرة

أوامر سير العمليات

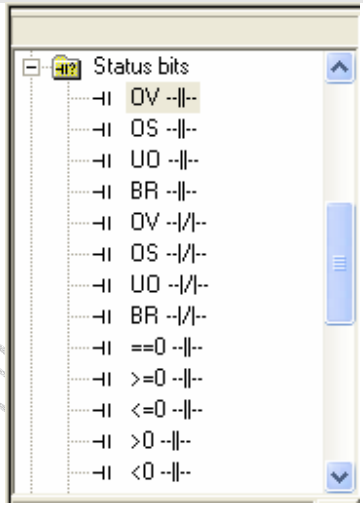
الأمر Sequence Control Relay(SCR)

وفيه يتم تتبع سير العمليات كبرنامج التشغيل والإيقاف وغيرها
وينقسم الأمر إلى عدة أقسام منها

Load Sequence Control Relay (LSCR)

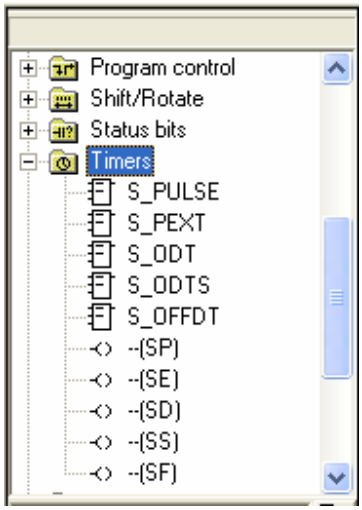
وفية يتم بداية تشغيل العملية
Sequence Control Relay Transition (SCRT)
وفيه يتم ترحيل العملية إلى المرحلة التي تليها
Sequence Control Relay End (SCRE)
وفيه يتم أنها العملية

أوامر المقارنة



تستخدم أوامر المقارنة لمقارنة الأعداد بجميع أنواعها وهى تعتبر من أهم الأوامر حيث تستخدم لمقارنة القيم القياسية لتنفيذ عمليات معينة ومن أهمها اكبر من اصغر من اكبر من أو يساوى اصغر من أو يساوى الخ.

أوامر المؤقتان

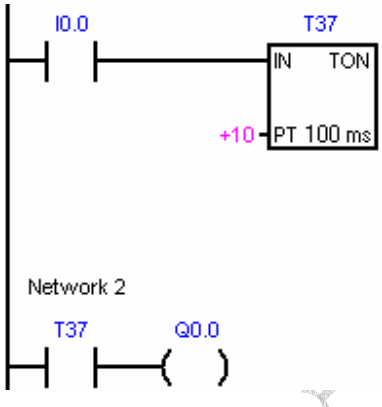


وهى من أهم أوامر لغات البرمجة وفى لغة البرمجة STEP7 يتم تقسيمها حسب نوعها ويتم تسميتها خلال البرنامج لاختيار زمن المؤقت والقائمة أدناه توضح أنواع الموقنات وأسمائها وأزمانها

Timer Type	Resolution	Maximum Value	Timer Number
TONR	1 ms	32.767 s	T0, T64
	10 ms	327.67 s	T1-T4, T65-T68
	100 ms	3276.7 s	T5-T31, T69-T95
TON, TOF	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33-T36, T97-T100
	100 ms	3276.7 s	T37-T63, T101-T255

مؤقت TON

وفيه يتم حساب قيمة المؤقت ثم عند انتهاء زمن العد تكون قيمة خرج المؤقت تساوى 1



في المثال أعلاه يوضح مؤقت من نوع TON بصفة T37 100ms حيث تم تعديل المؤقت PT=10 بمعنى 10*100=1 s

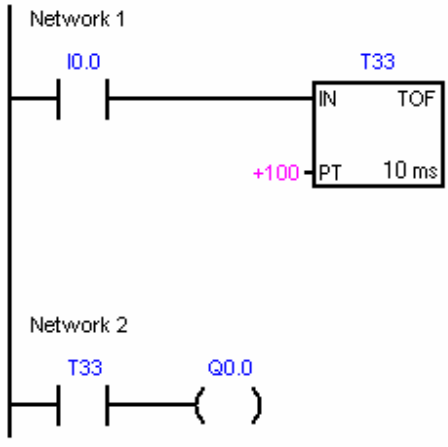
حيث يبدءا المؤقت بالعد عندما تكون قيمة إشارة الدخل I0.0 تساوى 1 وعند مرور 1 S تكون قيمة خرج المؤقت 1 مع ملاحظه انه يجب إن تكون قيمه الدخل I0.0 ثابتة اى تساوى 1

مؤقت نوع TOF

وفيه تكون قيمة خرج المؤقت تساوى 1 وعند إيقاف الموقت لا تتغير قيمة الخرج إلى 0 إلا بعد انتهاء زمن العد

المثال أدناه يوضح مؤقت من نوع TOF بصفة T33 حيث تم تعديل المؤقت PT=100 بمعنى 10*100=1 s

حيث عندما تكون قيمة إشارة الدخل I0.0 تساوى 1 تكون قيمة خرج المؤقت 1 وعند تغيير قيمة إشارة الدخل إلى 0 لاتتغير قيمة خرج المؤقت إلى 0 إلا بعد مرور 1S



مؤقت TONR

وهو شبيه بالمؤقت TON إلا انه يقوم بتخزين وجمع زمن العد إلى إن يطابق زمن التعديل

في المثال أدناه يوضح مؤقت من نوع TONR بصفة T1 بتعديل PT=100 اى 1S

حيث عندما تكون قيمة إشارة الدخل I0.0 تساوى 1 تكون قيمة خرج المؤقت تساوى 0 ونفرض إن بعد مرور 0.5S تغيرت قيمة إشارة الدخل I0.0 إلى 0 عندها يكون المؤقت قد خزن قيمة العد السابقة وإذا تغيرت قيمة إشارة الدخل I0.0 إلى 1 مره أخرى يقوم المؤقت بالعد 0.5 S عندها تكون قيمة عد المؤقت 1 S وفى هذه الحالة يكون قيمة خرج المؤقت تساوى 1