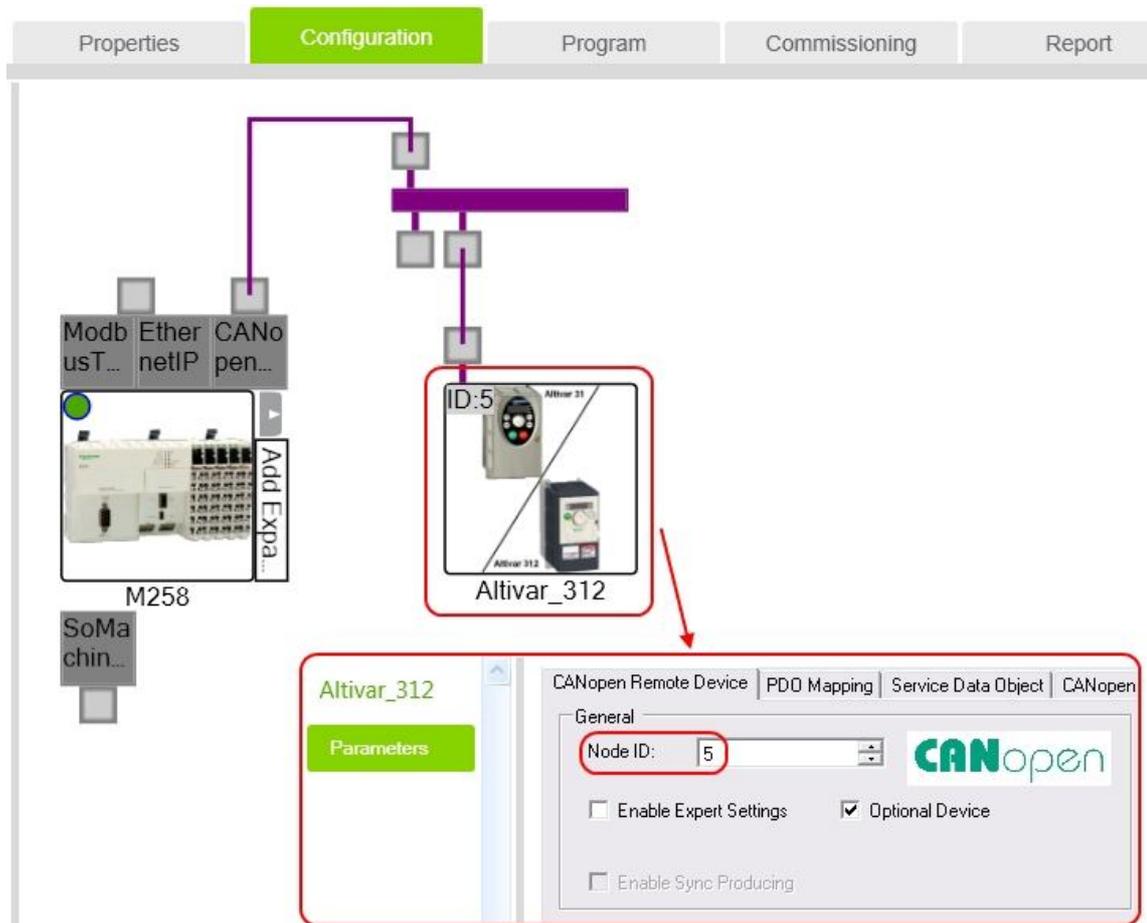


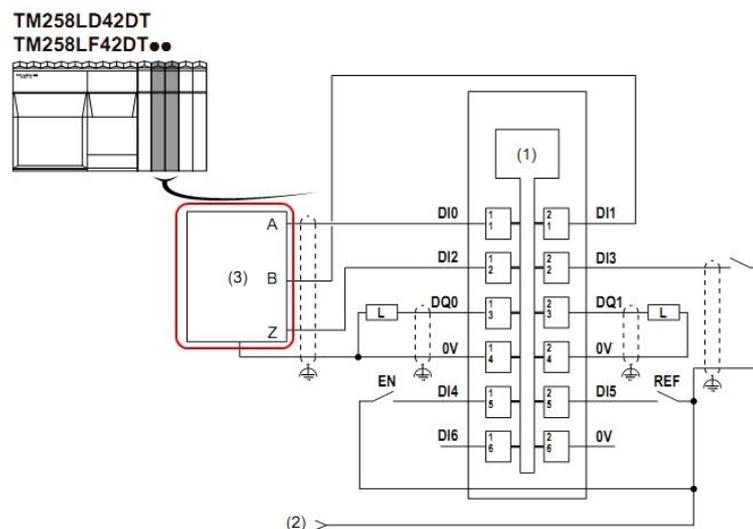
Упражнение 22. Работа с энкодером

1.1 Создайте в *SoMachine* новый проект с ПЛК *M258* и языком программирования *CFC* в программных модулях *POU*, назовите проект "22_1 Encoder".

1.2 Добавляйте в графическом конфигураторе шину *CANopen* к порту *CANopen* ПЛК, а затем к шине *CANopen* добавляйте ведомое устройство *Altivar 312*, настроив его сетевой адрес (на рисунке адрес ПЧ *Altivar 312* имеет значение «5», она должна отвечать настройкам ПЧ).

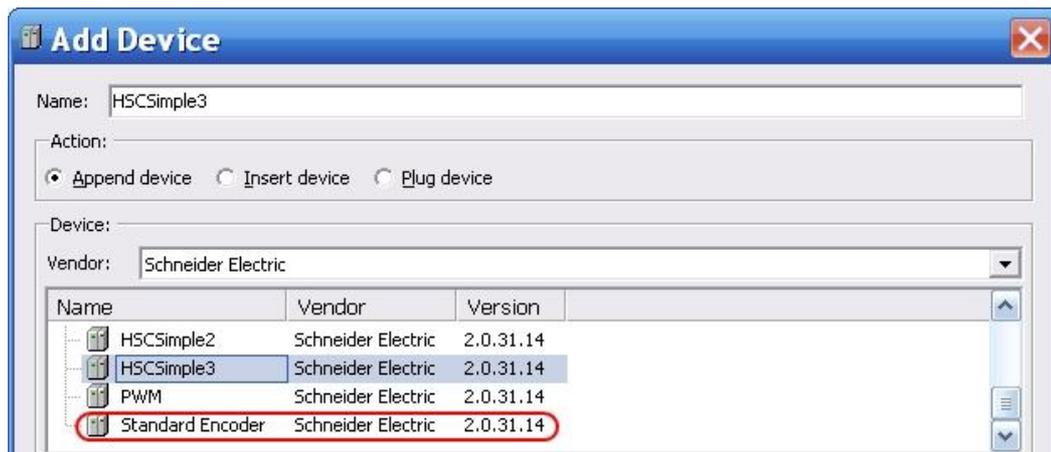


1.3 Проверьте подключение энкодера к быстрым входам ПЛК. На стенде используется подключение энкодера к входам I0 (A), I1 (B) и I2 (Z) встроенного модуля быстрого счета *DM72F0*. Подключение осуществляется по образцу в документации на аппаратное обеспечение ПЛК *M258*.

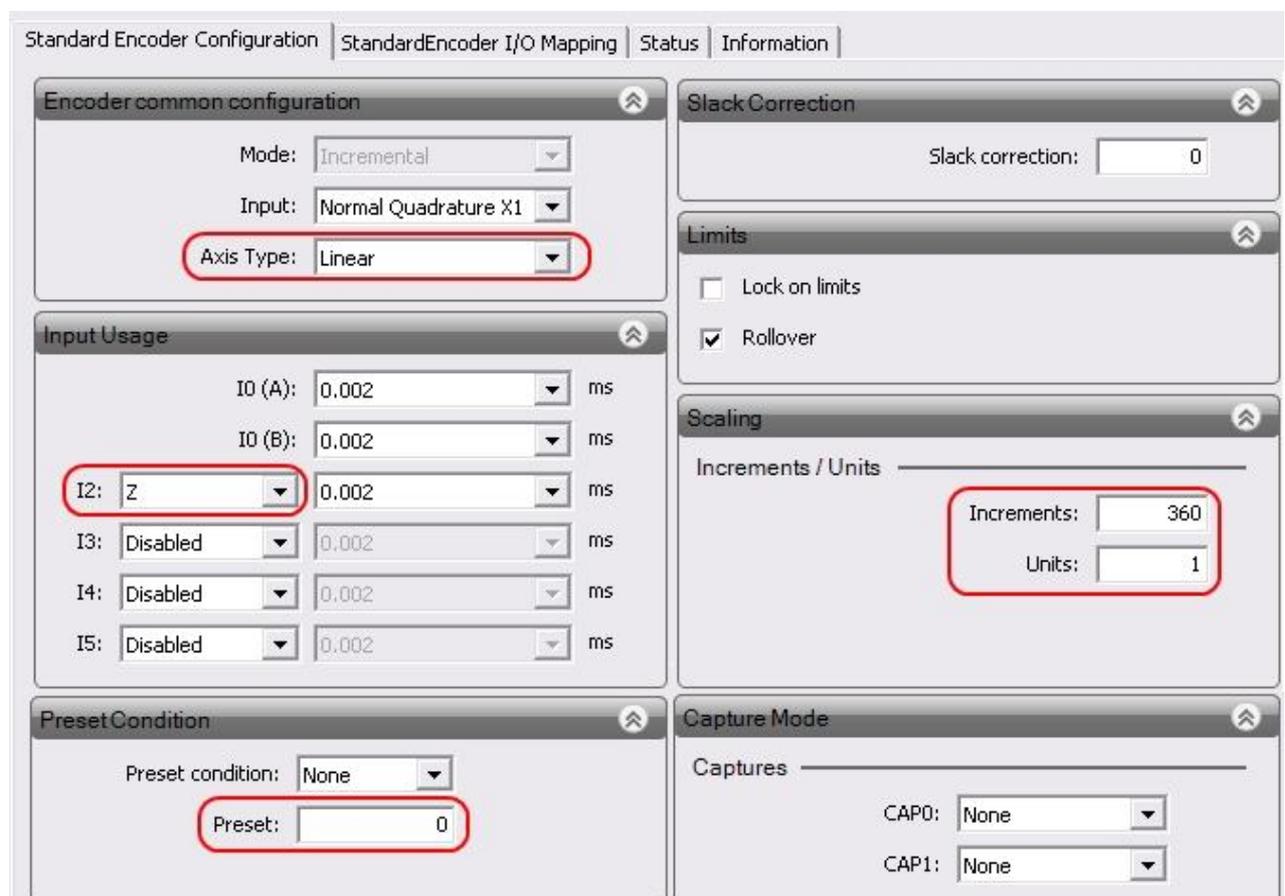


Вычисление скорости движения в проекте будет осуществляться с помощью функционального блока быстрого счетчика *HSC_Main*. Для подсчета частоты импульсов осуществлено подключение одного из выходов энкодера (например, A) к быстрому входу I0 другого встроенного модуля быстрого счета *DM72F1*.

1.4 Перейдите во вкладку *Program* среды *SoMachine*. Выберите из дерева проекта правым кликом встроенный модуль быстрого счета *Expert DM72F0* и добавьте в него устройство: стандартный энкодер (*Add Device / Standard Encoder*).



1.5 Перейдите в настройки стандартного инкрементного энкодера, сделав двойной клик на добавленном устройстве *Standard Encoder*. Настройте устройство в соответствии с рисунком.

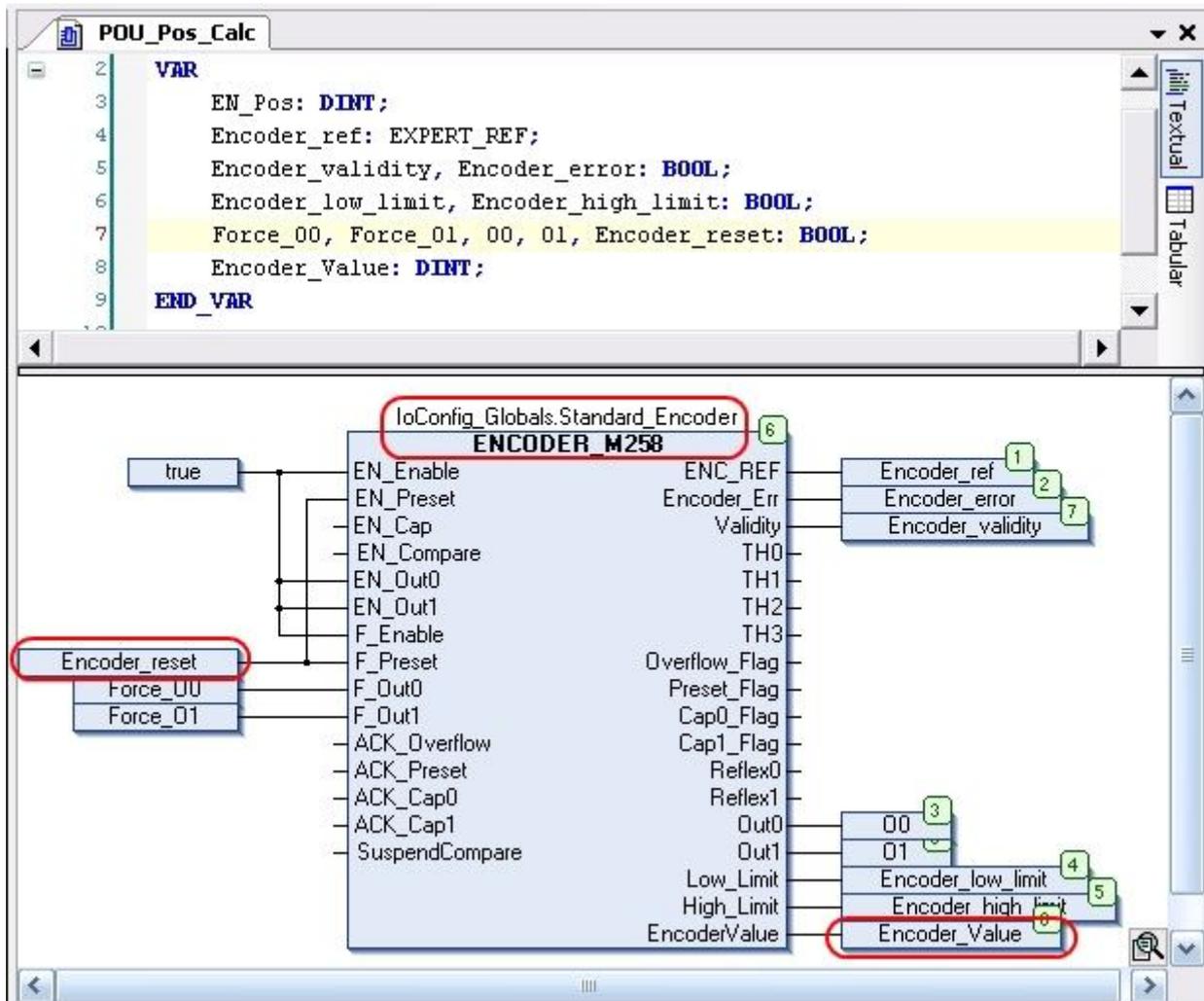


Настройки энкодера должны соответствовать задачам по определению положения вала оси и разрешению энкодера. В примере рассматривается энкодер с разрешением 360 точек на оборот, а линейное перемещение рабочего органа по оси составляет 1 ед. длины на 360 импульсов энкодера. Квадратура энкодера (X1, X2 или X4) позволяет увеличить точность определения положения за счет его вычисления по передним и задним фронтам импульсов на входах A и B (это нужно учитывать при вводе количества импульсов на единицу длины в поле *Increments*). Текущее значение положения вычисляется по формуле

$$\text{EncoderValue} = \frac{\text{Units}}{\text{Increments}} \cdot \text{CountPulses}.$$

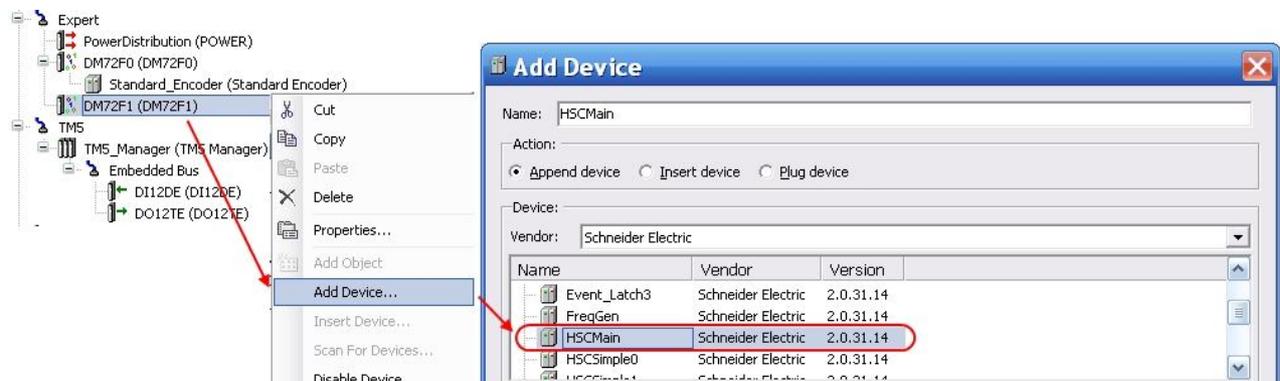
1.6 Для ветви *Application* дерева проекта создайте программный модуль *POU_Pos_calc* на языке программирования *CFC*. С помощью графического элемента *Box* создайте блок типа 22_1 Работа с энкодером

Encoder_M258 и присвойте ему экземпляр (имя блока) *IoConfig_Globals.Standard_Encoder*, используя помощник ввода. Привяжите входы / выходы блока *Encoder_M258* к переменным *POU* по указанному на рисунке примеру. Обратите внимание на типы переменных *Encoder_ref* и *Encoder_Value*.

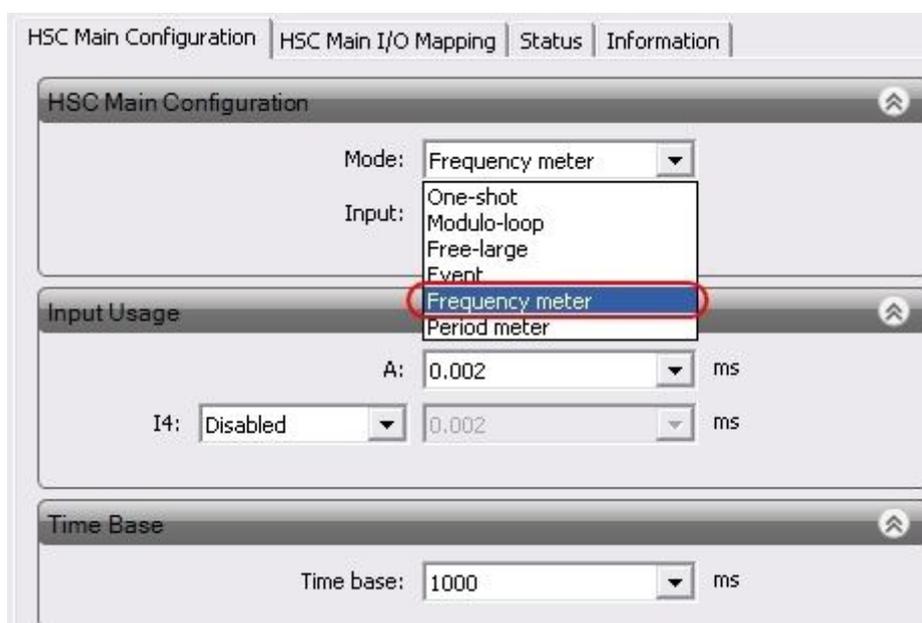


В указанном примере положение рабочего органа оси в единицах длины можно определять по значению переменной *Encoder_Value*. Сброс значения энкодера при настройке его нулевого положения в соответствии с положением вала механизма осуществляется по переднему фронту логической переменной *Encoder_Reset* (для этого в параметрах настройки стандартного энкодера *Standard Encoder Configuration* задайте значение уставки *Preset = 0*).

1.7 Для определения фактической скорости перемещения (вращения) воспользуемся быстрым счетчиком *HSCMain*, привязанным к модулю быстрого счета *DM72F1*. Добавьте указанный тип счетчика к модулю с помощью дерева проекта, выполнив команду *Add Device*.

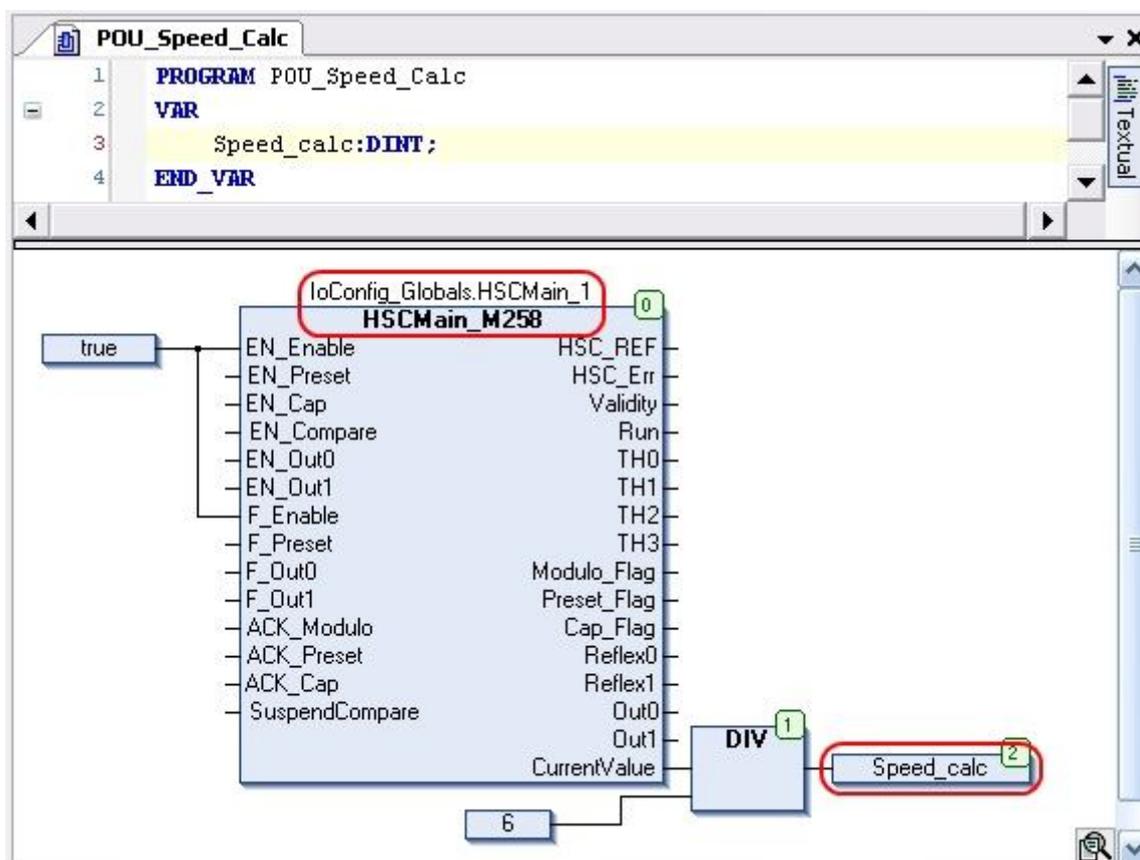


1.8 Откройте окно свойств быстрого счетчика типа *HSCMain* и настройте его тип на измерение частоты входных импульсов, используя одну из его функций *Frequency meter*. Настройте остальные элементы быстрого счетчика в соответствии с примером.



1.9 Для ветви *Application* дерева проекта создайте программный модуль *POU_Speed_calc* на языке программирования *CFC*. С помощью графического элемента *Box* создайте блок типа *HSCMain_M258* и присвойте ему экземпляр (имя блока) *IoConfig_Globals.HSCMain_1*, используя помощник ввода. Привяжите входы / выходы блока *HSCMain_M258* к переменным *POU* по указанному на рисунке примеру. Обратите внимание на тип переменной, связанной с выходом функционального блока *CurrentValue*. Значение, получаемое на указанном выходе, является частотой импульсов на быстром входе в Гц.

Для перевода измеренной частоты в необходимое значение скорости (линейной или окружной) используйте блоки деления (умножения) в соответствии с задачей. В указанном примере один оборот вала оси соответствует 360 импульсов на входе быстрого счетчика, если за секунду вал будет делать 25 оборотов (1500 об/мин), это будет соответствовать частоте на входе быстрого счетчика $25 \times 360 = 9000$ Гц. Для перевода указанной частоты в значение окружной скорости (об/мин) в примере используется деление *CurrentValue* на 6: $9000 \text{ (Гц)} / 6 = 1500 \text{ (об/мин)}$.

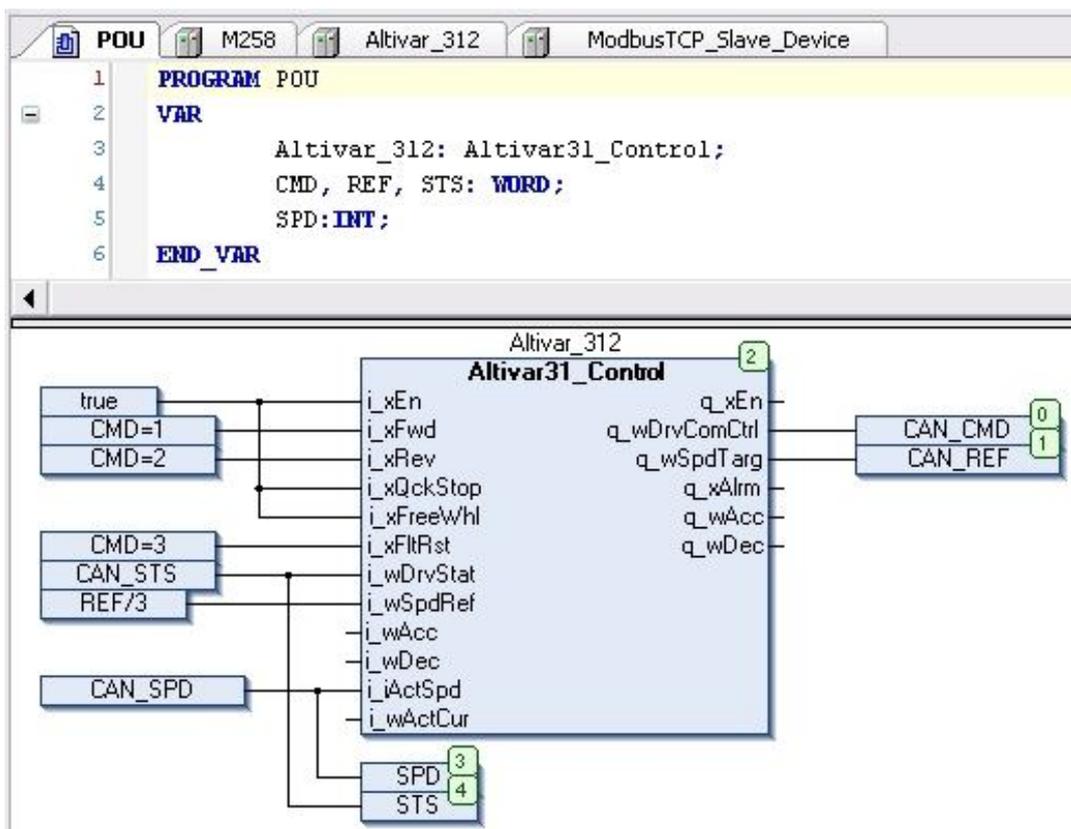


1.10 Создайте приложение для управления преобразователем частоты *Altivar 312*. Из бокового браузера проекта выберите в устройствах *CANopen* ПЧ *Altivar_312* и откройте вкладку 22_1 Работа с энкодером

настроек переменных чтения / записи *CANopen I/O Mapping*. Привяжите переменные *Drivcom command reg.* (слово управления шины CANopen), *Frequency Reference* (задание частоты по шине CANopen в 0,1 Гц), *Drivcom status register* (слово статуса ПЧ) и *Control Effort* (фактическая скорость вращения оси) к переменным структуры оси вращения *Altivar_312 CAN_CMD*, *CAN_REF*, *CAN_STS* и *CAN_SPD* соответственно. Эти переменные будут использоваться в программном модуле *POU*.

Variable	Mapping	Channel	Address	Type	Default Va...	Unit
CAN_CMD		Drivcom command reg.	%QW24	UINT	0	
CAN_REF		Frequency Reference	%QW25	INT	0	
CAN_STS		Drivcom status register	%IW27	UINT	0	
		Extended status register	%IW28	UINT	0	
CAN_SPD		Control effort	%IW29	INT	0	
		Logic Input/Output image	%IW30	UINT	0	

1.11 Используя панель инструментов среды *CFC SoMachine* разработайте программу управления преобразователем частоты. Образец программы с использованием функционального блока *Altivar31_Control* библиотеки *Schneider Electric* приведен ниже. Значения слова управления *CMD* следующие: 0 – остановка, готовность; 1 – движение вперед; 2 – движение назад; 3 – сброс неисправности. Для задания скорости вращения в об/мин нужно учитывать паспортные данные двигателя (количество пар полюсов).



1.12 Дополните программу определения скорости перемещения (вращения) *POU_Speed_calc* необходимыми функциональными блоками для определения знака скорости (вращение вперед или назад) *самостоятельно*, используя в качестве данных частоту на выходе преобразователя *SPD* или значение энкодера *EncoderValue*.

1.13 Откомпилируйте проект, сохраните и загрузите его в ПЛК. Проверьте работу проекта в режиме он-лайн.

MyController.Application.POU_Pos_Calc

Expression	Type	Value	Prepared value	Comment
O1	BOOL	FALSE		
Encoder_reset	BOOL	FALSE		
Encoder_value	DINT	265		

MyController.Application.POU_Speed_Calc

Expression	Type	Value	Prepared value	Comment
Speed_calc	DINT	1227		

1.14 Сохраните проект, создав его резервную копию. Сделайте выводы по работе.