



## **Контроллеры промышленные серии AX-3**

**Описание примера  
«AX-308E\_Pulse\_Axis»**

**[deltronics.ru](http://deltronics.ru)**

## Оглавление

Тема примера .....	3
Настройки сервопривода Delta ASD-E3-L .....	4
Настройки в проекте контроллера .....	7

## Тема примера

Как создать в проекте DIADesignerAX импульсную ось для контроллеров серии AX-308E/316E

Данный пример рассматривает основные вопросы организации импульсной оси позиционирования. Также рассматривается процедура использования счётчика в качестве обратной связи от энкодера двигателя для контроля позиции.

Основной (запускающий) файл примера проекта для контроллера AX-308E:

**AX-308E\_Pulse\_Axis.project**

Среда программирования DIADesignerAX версии не ниже 1.5.0

Состав оборудования: контроллер AX-308E и сервопривод Delta ASD-E3-L с импульсным управлением

## Настройки сервопривода Delta ASD-E3-L

Перед началом настройки сервопривода его необходимо сбросить на заводские настройки. Для этого введите число 10 в параметр P2.008 и перегрузить привод по питанию.

Введите следующие параметры:

P1.000 = 0x1040 (4AB, positive logic, filter 1000 kHz)

P1.001 = 0x0000 (режим PT, импульсное задание)

Настройки числителя/знаменателя привода, чтобы для 1 оборота вала нужно было подать ровно 10000 импульсов:

P1.044 = 16777216

P1.045 = 10000

Настройка количества выдаваемых импульсов на 1 оборот вала:

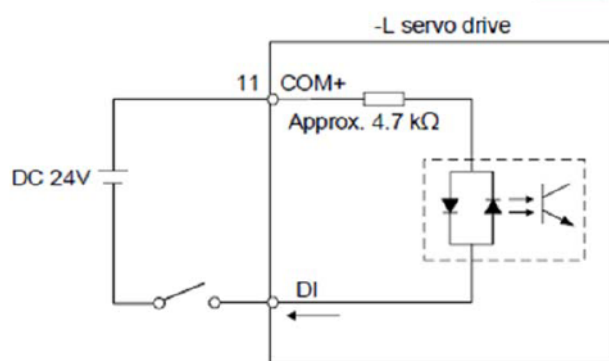
P1.046 = 2500

P2.010 = 0x0101 (вход DI1 = Servo-ON)

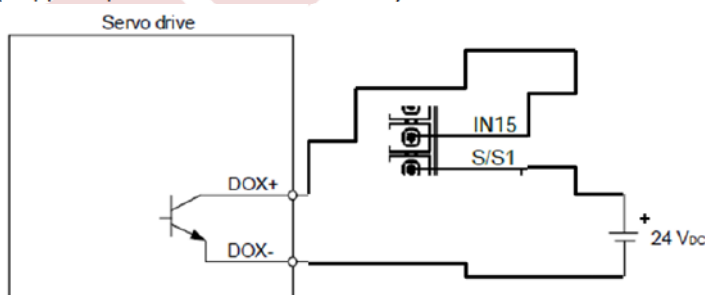
P2.011 - P2.017 = 0 (с остальных входов функции сняты)

P2.018 = 0x0102 (выход DO1 = Servo-ON)

Физически выход контроллера Q07 соединён с входом DI1 сервопривода по следующей схеме: (команда Servo-ON)

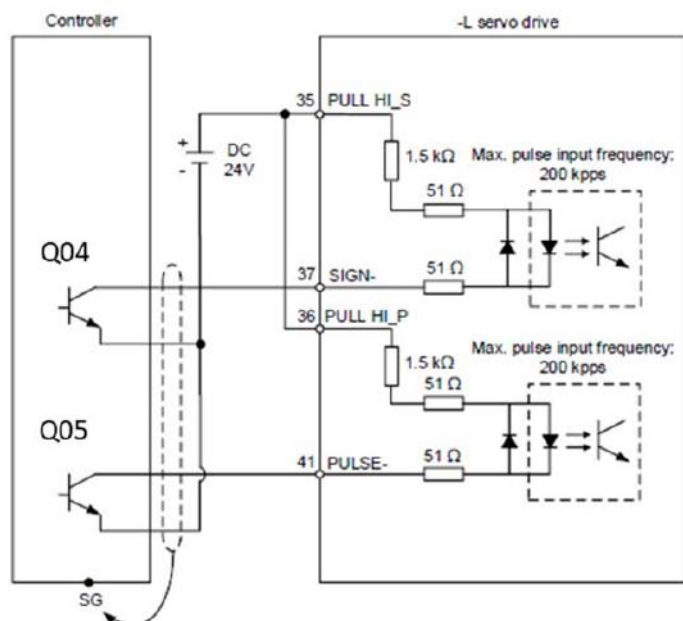


Выход сервопривода DO1 соединён со входом контроллера IN15 по следующей схеме: (индикация состояния Servo-ON)



6	DO1-	Digital output
7	DO1+	Digital output

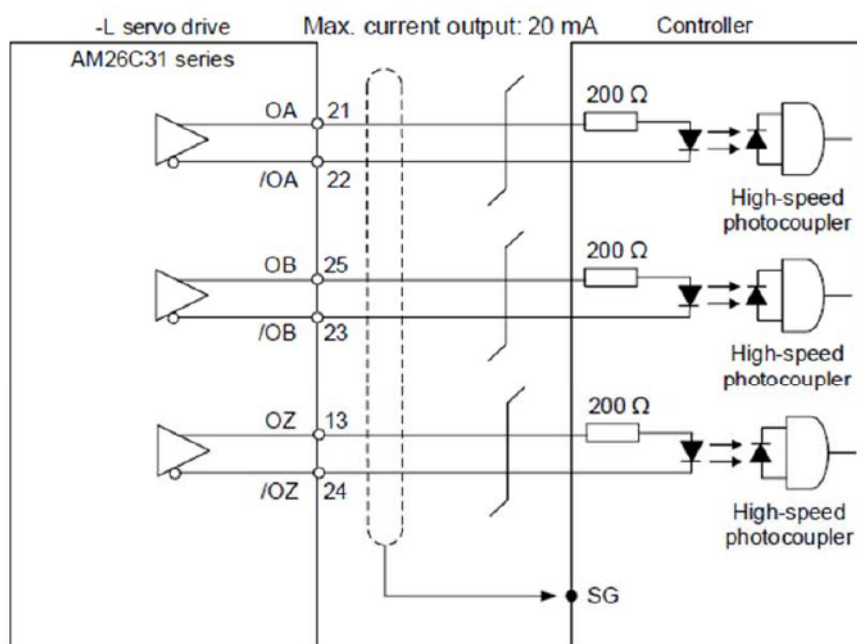
Выдача импульсов на привод осуществляется с выходов Q04/Q05, которые соединены с импульсными входами сервопривода по следующей схеме:



Импульсные выходы энкодера (фазы А и В) привода заводятся на входы энкодера в разъёме DB15 контроллера по следующим схемам:

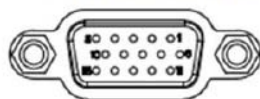
Привод, контакты разъёма CN1 (21/22 и 25/23). Тип сигнала 5 V TTL

C10: output for encoder position signal (photocoupler)



Клеммы на разъёме энкодера контроллера. Тип сигнала 5 V TTL.

Encoder interface



Encoder



Pin	Function
1	A1+
2	A1-
10	B1+
11	B1-
4	Z1+
5	Z1-
15	5V
8	GND

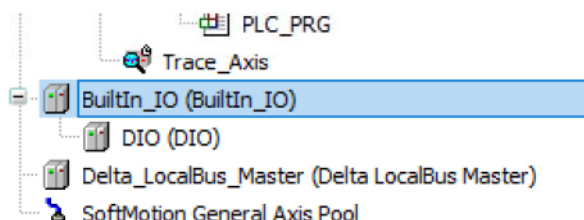
Function
A
$\bar{A}$
B
$\bar{B}$
Z
$\bar{Z}$
VCC
0V



## Настройки в проекте контроллера

Среда программирования для контроллеров серий AX-308E/316E содержит удобную процедуру настройки импульсной оси движения.

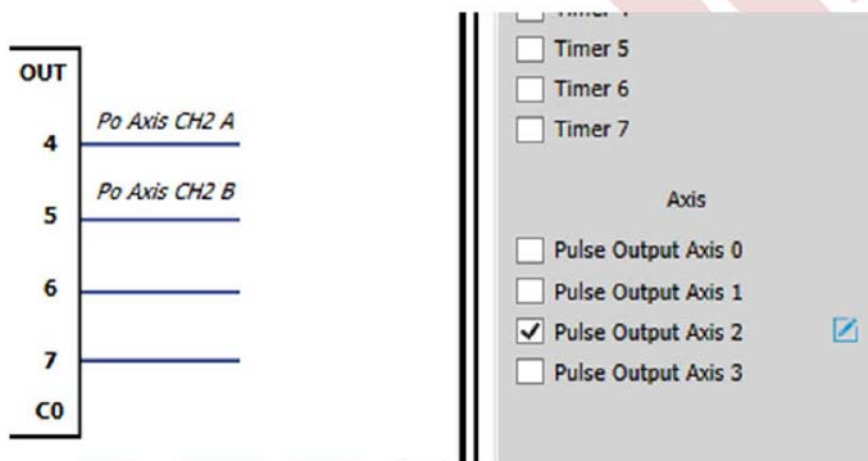
Найдите в древе проекта пункт **BuiltIn\_IO** и щёлкните на нём два раза левой кнопкой мышки.



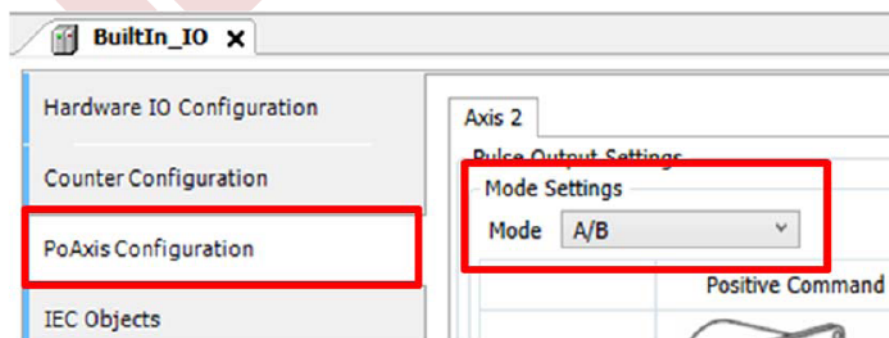
В открывшемся окне выберите пункт **Hardware IO Configuration**:



Поставьте флажок на нужной оси. В нашем примере это ось **Axis 2**. Мастер автоматически зарезервирует под неё выхода 4 и 5:



Далее в разделе **PoAxisConfiguration** установите режим (Mode) как **A/B** :



Количество импульсов на 1 оборот двигателя как 10000

Mechanism Settings

(1) Command pulse per motor rotation: 10000 [ Pulse ]

(4) Pitch: 1 [ Unit ]

Линейное перемещение на 1 оборот двигателя 10 единиц (любые единицы пользователя).  
(данный параметр используется потом во всех инструкциях движения. Т.е. при задании перемещения в 10 единиц, мотор сделает 1 оборот).

Mechanism Settings

(1) Command pulse per motor rotation: 10000 [ Pulse ]

(4) Pitch: 10 [ Unit ]

Название оси, которое необходимо указывать в инструкциях движения можно увидеть в пункте IEC Objects:

Hardware IO Configuration	Variable	Type	Configuration Function
Counter Configuration	Counter_0	DFB_COUNTER_REF	Counter 0
PoAxis Configuration	Encoder_Axis	DMC_ENCODER_AXIS_REF	Counter 0/FreeEncoder_Axis
IEC Objects	Pulse_Output_Axis_2	DMC_PULSE_AXIS_REF	Pulse Output Axis 2

Необходимо иметь ввиду, что в инструкциях движения в единицах пользователя задаётся не только расстояние, но и скорость, а также ускорение.

Расчёт единиц скорости и ускорения осуществляется следующим образом:

Предположим, что конечный механизм проходит 10 мм на один оборот двигателя.

Для инструкций движения:

Задание положения в 10 ед. = 1 оборот мотора (конечный механизм пройдёт 10 мм)

Максимальная скорость:  $3000 : 60 = 50$  об/сек;  $50 * 10$  ед. расстояния = 500 ед. расстояния в сек  
(т.е. механизм пройдёт 500 мм за 1 секунду, и это максимальная скорость, выраженная через единицы расстояния в секунду)

Т.е. 3000 об/мин = 500 ед./сек

Ускорение. Предельно возможное ускорение до 3000 об/мин за 1 мс: 50 об/сек надо умножить на 1000, получим приращение скорости (ускорение) за 1 мс :  $50 * 1000 = 50000$  ед. расстояния за 1 мс.

Итого:

50000 = ускорение до 3000 об/мин за 1 мс (предельное теоретическое ускорение)

500 = ускорение до 3000 об/мин за 100 мс

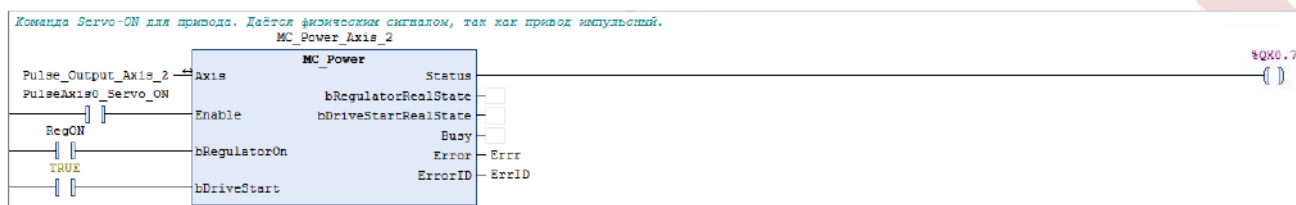
50 = ускорение до 3000 об/мин за 1 с

(т.е. 50 ед. ускорения - это ускорение до скорости 50 об/сек за 1 секунду).

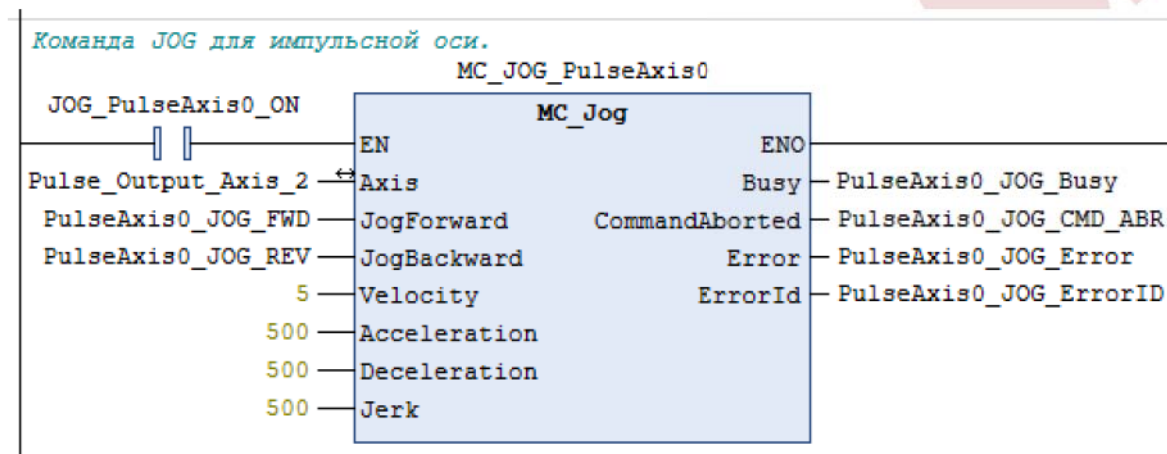


Импульсные выходы контроллера имеют частоту 200 кГц. При настройке 10000 импульсов, привод сможет вращаться с максимальной скоростью  $200\,000 : 10\,000 = 20$  об/сек, или  $20 * 60 = 1200$  об/мин. Однако, в приводе используется режим 4AB, что позволяет считать все фронты отдельно. Т.е. входная частота умножается на 4. Таким образом, можно достигнуть скорости в 3000 об/мин (4800 макс). Но надо помнить, что скорость и расстояние в инструкциях движения нужно задавать в 4 раза меньше нужной.

Поскольку привод управляется физическими сигналами, то команду Servo-ON и её подтверждение необходимо осуществлять через физические входы-выходы:

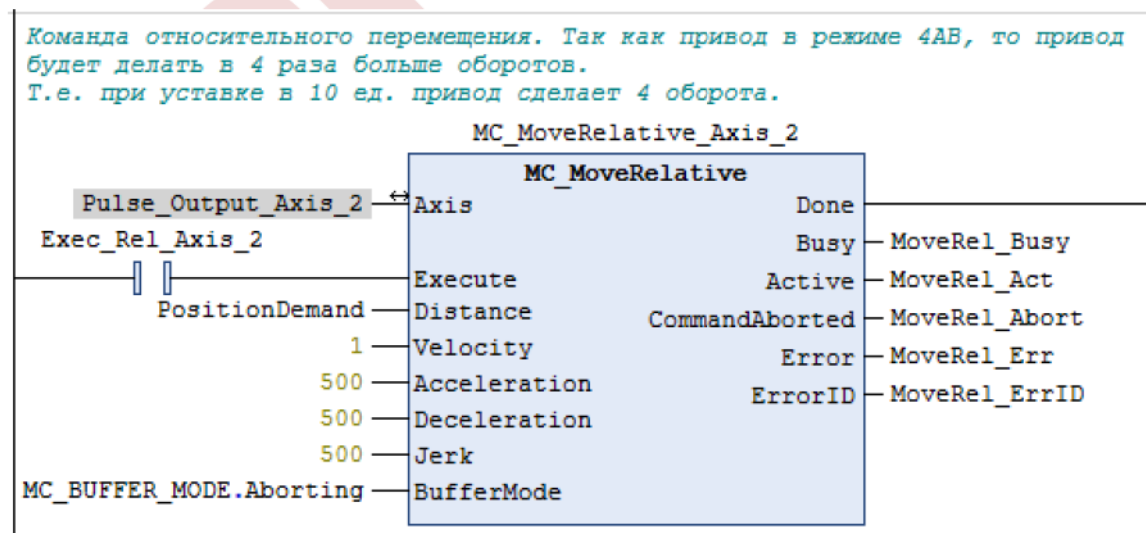


### Пример команды JOG



Обратите внимание, что задание скорости установлено в 5 ед./сек. При режиме pulse/dir это будет соответствовать скорости 30 об/мин, но так как у нас стоит режим 4AB, то скорость привода будет 120 об/мин.

### Пример команды MoveRelative



Так как привод в режиме 4AB, то привод будет делать в 4 раза больше оборотов.  
Т.е. при уставке в 10 ед. привод сделает 4 оборота, хотя в данном примере стоит 10 ед. на 1 оборот. Но так как привод считает отдельно все 4 фронта AB, то получается 4 оборота.

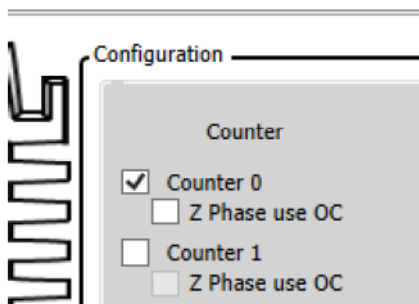
Для подсчёта импульсов от энкодера двигателя с целью контроля реального положения можно использовать счётчик с уровнем сигнала 5 V TTL. У контроллеров AX-308E/316E это счётчики Counter\_0 и Counter\_1 в разъёме DB15. В нашем примере используется Counter\_0. Схема подключения приведена выше в разделе настройки привода.

Среда программирования DesignerAX имеет для контроллеров AX-308E/316E удобную форму настройки высокоскоростных счётчиков.

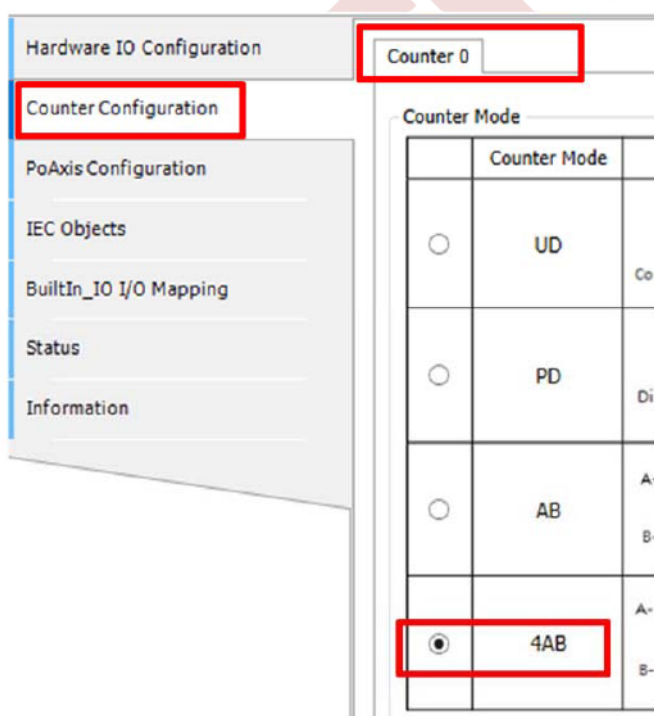
В вкладке **BuiltIn\_IO** выберите пункт **Hardware IO Configuration**:



Отметьте счётчик Counter\_0



В пункте Counter Configuration выберите режим 4AB

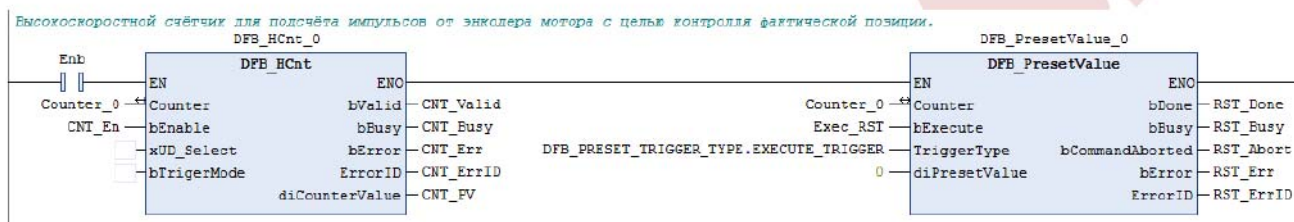


Параметр привода P1.046 = 2500, следовательно в режиме 4AB счётчик будет подсчитывать 10000 импульсов на 1 оборот вала мотора.

Имя счётчика можно увидеть в разделе IEC Objects:

Hardware IO Configuration	Variable	Type	Configuration Function
Counter Configuration	Counter_0	DFB_COUNTER_REF	Counter 0
PoAxis Configuration	Encoder_Axis	DMC_ENCODER_AXIS_REF	Counter 0/FreeEncoder_Axis
IEC Objects	Pulse_Output_Axis_2	DMC_PULSE_AXIS_REF	Pulse Output Axis 2
BuiltIn_IO I/O Mapping			

В программе для получение числа подсчитанных импульсов используется команда DFB\_HCnt, а для сброса счётчика команда DFB\_PresetValue с нулевой уставкой:



Также, командой MC\_ReadActualPosition можно считывать позицию импульсной оси накопительным итогом. Фактически это виртуальная ось, поэтому необходимо понимать, что с реальным значением в регистре текущей позиции привода это никак не связано. Но для программных целей использовать удобно. Сбросить виртуальную ось можно командой MC\_SetPosition с нулевой уставкой. Но нужно опять же понимать, что это не сбрасывает текущую позицию в самом приводе. Но так как импульсная ось по своей природе всё равно виртуальная, то для программных целей в данном случае это будет корректно. Для сброса текущей позиции в приводе нужно выполнить выход в ноль в соответствии с процедурой привода, и это опять же не будет связано с позицией в программе.

