Input Mode – Joystick

Included programs:

Position Input Mode.qcp
Velocity Input Mode.qcp
Torque Input Mode.qcp

Overview

Position, velocity and torque can be continuously controlled from a single, raw data source using the Input Mode commands:

- Position Input Mode (PIM)
- Velocity Input Mode (VIM)
- Torque Input Mode (TIM)

The raw data is called Input Source Data and can come from many sources including an analog input, a host, a CANOpen® register map, an internal calculations, etc…

The Input Mode commands allow the user to filter, clip and scale the Input Source Data to create a smooth, well behaved output. For example, let us say the user wants to use a joystick to command the servo to go from –2000 counts to +2000 counts. The PIM command will allow the user to:

- Filter Out High Frequency Analog Input Noise
- Choose Amount of Joystick Travel To Use
- Define a Dead Band Around the Joystick Center Position
- Define Number of Counts to Move
- Limit Move Velocity

This application note describes how to configure the Input Mode commands for the above mentioned options and in particular gives instructions on how to use a joystick to command position, velocity or torque.

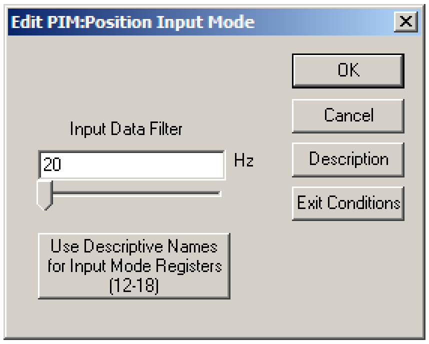
Edit Dialog Box

The Input Mode commands use registers 12-18 and command parameters to configure all of its options. They share a common edit dialog box (PIM shown below).

The Input Mode commands are modal. That is, once executed, they will continuously process the Input Source Data until a predefined exit condition is met or the Input Mode command is overridden.

CANopen® and CiA® are registered community trade marks of CAN in Automation e.V.

QuickControl® and QCI® are Registered Trademarks of QuickSilver Controls, Inc.
SilverLode™, SilverNugget™, SilverDust™, PVIA™, QuickSilver Controls™, and AntiHunt™ are trademarks of QuickSilver Controls, Inc..


Edit PIM:Position Input Mode

Input Data Filter

20 Hz

Input Data Filter

This is a low pass filter on the Input Source Data. Simply put, Input Source Data changing slower than the specified frequency is "passed", while higher frequency data is considered noise and is ignored. The lower this number the slower the Input Mode command will respond.

Use Descriptive Names

Changes the a names of registers 12-18 to their defaults

Exit Conditions

Press this button to select which conditions will cause the Input Mode command to exit.

Input Mode Registers 12-18

Input Mode commands use registers 12-18, which must be initialized before executing the command. After the Input Mode command is executed, the registers are continuously read so changes to any register will affect the motion in real-time.

Registers 12-18 are defined as follows:

Register	Description	Range
12	Input Source Data	-2,147,483,648 to +2,147,483,647 (1)
13	Input Offset	-2,147,483,648 to +2,147,483,647 (1)
14	Input Dead Band	0 to 32767
15	Maximum Scale/Limit	0 to 32767
16	Maximum Output Scale	-2,147,483,648 to +2,147,483,647
17	Output Offset	-2,147,483,648 to +2,147,483,647
18	Output Rate of Change	0 to +2,147,483,647

(1) Although register 12 and 13 may contain the indicated range, the Input Mode commands can at most process data in register 12 that ranges +/- 32767 from register 13.
(Reg 13 – 32767) <= Reg 12 <= (Reg 13 + 32767).

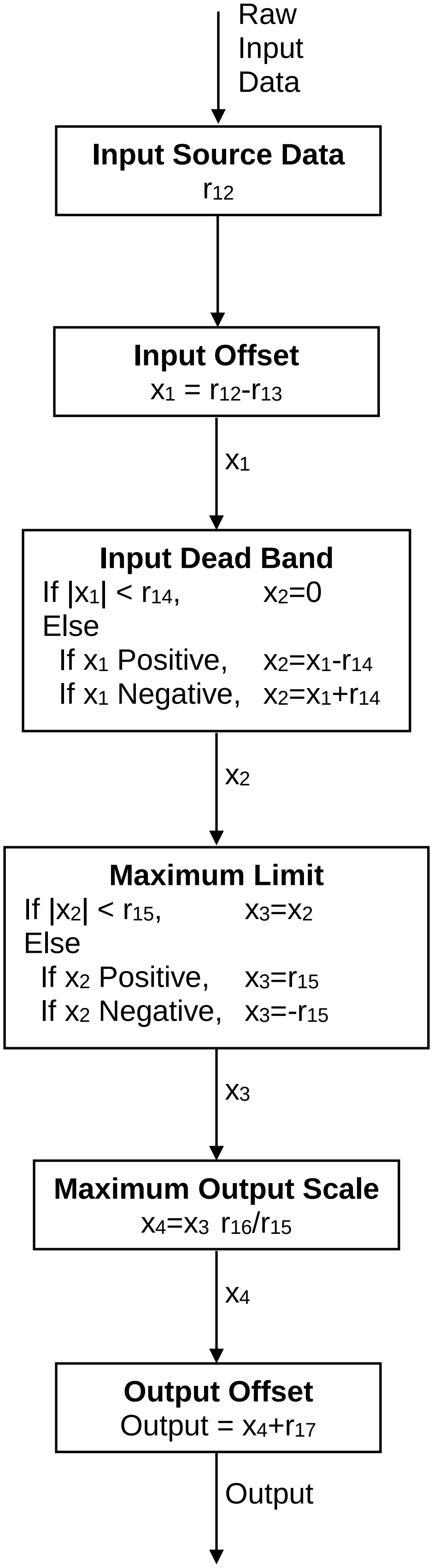
The Input Mode commands take the Input Source Data from register 12 and use registers 13-17 to sequentially process it through the steps shown in the Input Mode Flow Diagram (see following page).

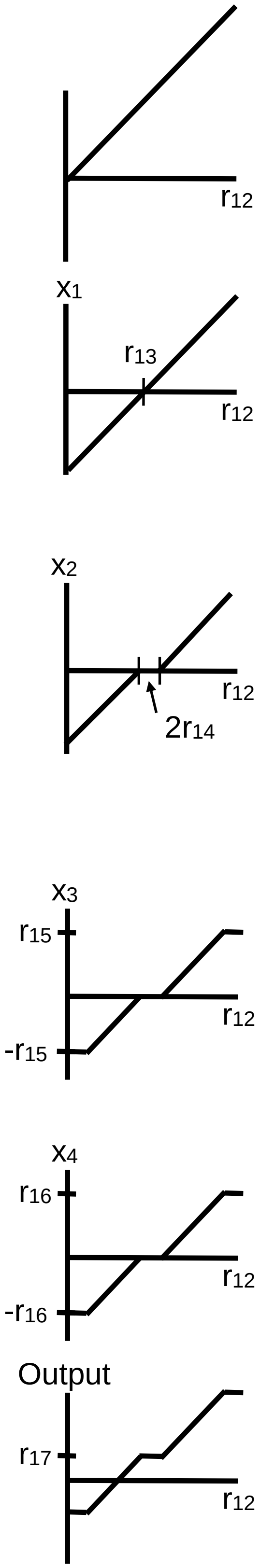
Register 18 defines the outputs maximum rate of change. For example, the register 18 defines the maximum commanded velocity for the PIM command the maximum command acceleration for the VIM command.

Legend

r_n: Register n

x_n: Intermediate Term n





Register 13 establishes the “Zero Cross Over”

Register 14 defines a dead band around the Zero Cross Over. Changes within this area do not result in motion.

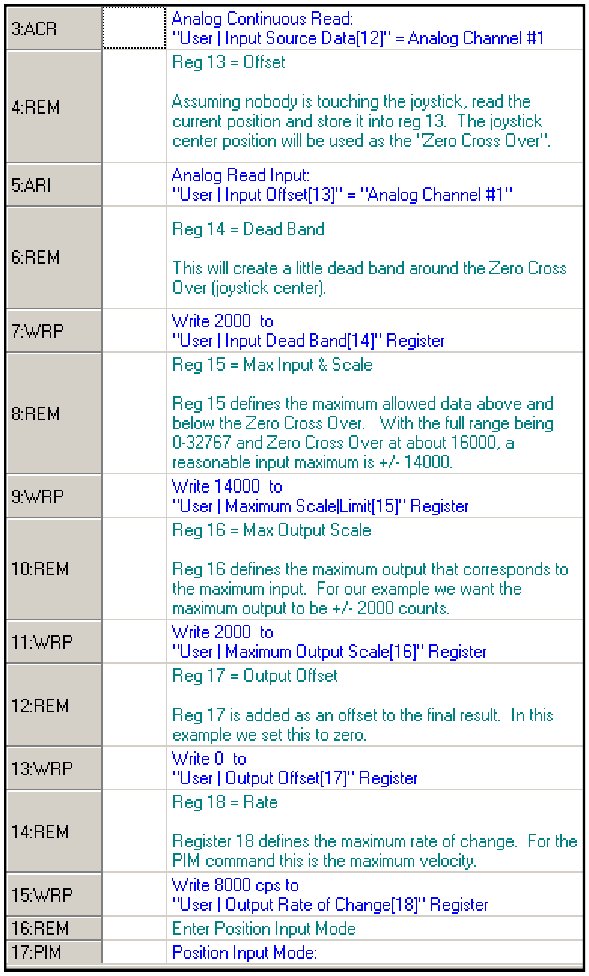
Register 15 defines the maximum allowed data above and below the Zero Cross Over.

Register 16 defines the maximum output that corresponds to the maximum input.

Register 17 is added as an offset to the output

Position Input Mode.qcp

Assume a Joystick is wired to Analog Input #1. This example will demonstrate how to configure Position Input Mode (PIM) to make the servo move +/- 2000 counts as the joystick moves.



3:ACR		Analog Continuous Read: "User | Input Source Data[12]" = Analog Channel #1
4:REM		Reg 13 = Offset Assuming nobody is touching the joystick, read the current position and store it into reg 13. The joystick center position will be used as the 'Zero Cross Over".
5:ARI		Analog Read Input: "User | Input Offset[13]" = "Analog Channel #1"
6:REM		Reg 14 = Dead Band This will create a little dead band around the Zero Cross Over (joystick center).
7:WRP		Write 2000 to "User | Input Dead Band[14]" Register
8:REM		Reg 15 = Max Input & Scale Reg 15 defines the maximum allowed data above and below the Zero Cross Over. With the full range being 0-32767 and Zero Cross Over at about 16000, a reasonable input maximum is +/- 14000.
9:WRP		Write 14000 to "User | Maximum Scale|Limit[15]" Register
10:REM		Reg 16 = Max Output Scale Reg 16 defines the maximum output that corresponds to the maximum input. For our example we want the maximum output to be +/- 2000 counts.
11:WRP		Write 2000 to "User | Maximum Output Scale[16]" Register
12:REM		Reg 17 = Output Offset Reg 17 is added as an offset to the final result. In this example we set this to zero.
13:WRP		Write 0 to "User | Output Offset[17]" Register
14:REM		Reg 18 = Rate Register 18 defines the maximum rate of change. For the PIM command this is the maximum velocity.
15:WRP		Write 8000 cps to "User | Output Rate of Change[18]" Register
16:REM		Enter Position Input Mode
17:PIM		Position Input Mode:

Line 3: Use Analog Continuous Read (ACR) to continuously read the joystick into register 12. Once executed, ACR continues in the background.

ACR takes the 10 bit analog input and scales it to a 15 bit number

SilverNugget: 0 to 5V == 0 to 32767
SilverDust: 0 to 3.3V == 0 to 32767

For details on analog inputs, see Application Note QCI-AN023 Analog Inputs.

Line 5: Assuming nobody is touching the joystick, read the joystick's center position into the Input Offset register (reg 13). The joystick center position will be used as the "Zero Cross Over". If the joystick is in the center it will read about 16000.

Line 7: A 2000 count dead band is established around the Zero Cross Over to give the joystick's center position a range of no motion.

Line 9: Setting the Maximum Input|Scale register (reg 15) to 14000 and assuming the Zero Cross Over is 16000, limits the input range between 2000 and 30000 thus clipping the joystick ends which are typically different from one joystick to another.

Line 11: Register 16 defines the maximum output that corresponds to the maximum input. For our example we want the maximum output of +/- 2000 counts to correspond to the input range of +/- 14000 around Zero Cross Over.

Line 13: For this example we set the Output Offset to 0. If, instead, you wanted the output to range between –1000 and +3000, you would set this register to 1000.

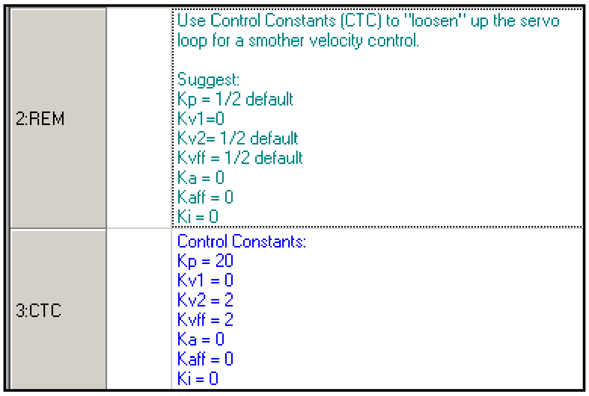
Line 15: Register 18 is set to 8000cps (native value=32212255) to limit the commanded velocity.

Use Register Watch to monitor register 12 as the joystick moves back and forth. Note, the dead band and clipped end of travel. See the effects of registers 13-18 by changing them in Register Watch.

Velocity Input Mode.qcp

Assume a Joystick is wired to Analog Input #1. This example will demonstrate how to configure Velocity Input Mode (VIM) to make the servo move +/- 20000cps as the joystick moves from left to right.

Line 3: Control Constants (CTC) are modified as shown to “loosen” up the servo loop. When using a joystick to command velocity, a smooth velocity is usually more important than aggressive position control.



2:REM	Use Control Constants (CTC) to "loosen" up the servo loop for a smother velocity control.
	Suggest:
	Kp = 1/2 default
	Kv1=0
	Kv2= 1/2 default
	Kvff = 1/2 default
	Ka = 0
	Kaff = 0
	Ki = 0

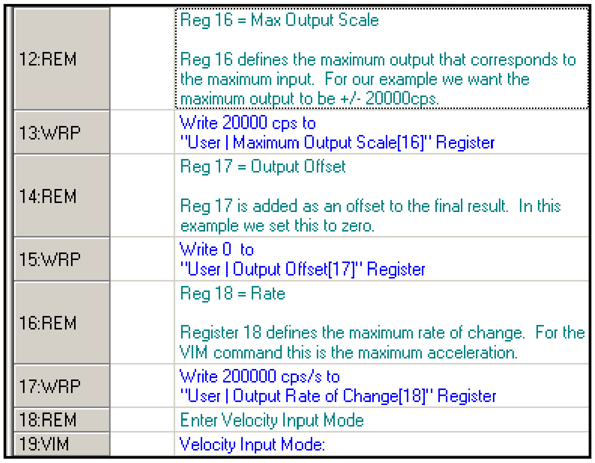
3:CTC	Control Constants:
	Kp = 20
	Kv1 = 0
	Kv2 = 2
	Kvff = 2
	Ka = 0
	Kaff = 0
	Ki = 0

Lines 5-11: Registers 12-15 have the same setup as the above Position Input Mode.qcp example.

Line 13: Register 16 defines the maximum output that corresponds to the maximum input. For our example we want the maximum output to be +/- 20,000cps (native value=80530637).

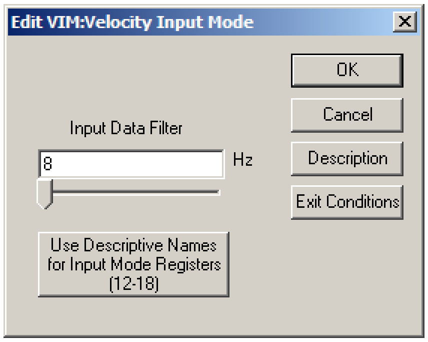
Line 15: For this example we set the Output Offset to 0. If, instead, you wanted the joystick to modify a base speed, you would set register 17 to the base speed.

Line 17: Register 18 is set to limit the commanded acceleration.



12:REM	Reg 16 = Max Output Scale Reg 16 defines the maximum output that corresponds to the maximum input. For our example we want the maximum output to be +/- 20000cps.
13:WRP	Write 20000 cps to "User | Maximum Output Scale[16]" Register
14:REM	Reg 17 = Output Offset Reg 17 is added as an offset to the final result. In this example we set this to zero.
15:WRP	Write 0 to "User | Output Offset[17]" Register
16:REM	Reg 18 = Rate Register 18 defines the maximum rate of change. For the VIM command this is the maximum acceleration.
17:WRP	Write 200000 cps/s to "User | Output Rate of Change[18]" Register
18:REM	Enter Velocity Input Mode
19:VIM	Velocity Input Mode:

Line 19: The Input Data Filter is set to 8Hz to filter out quick joystick movements, thus smoothing the input data.



Edit VIM:Velocity Input Mode

Input Data Filter: 8 Hz

Torque Input Mode.qcp

Assume a Joystick is wired to Analog Input #1. This example will demonstrate how to configure Torque Input Mode (TIM) to make the servo move +/- as the joystick moves from left to right.

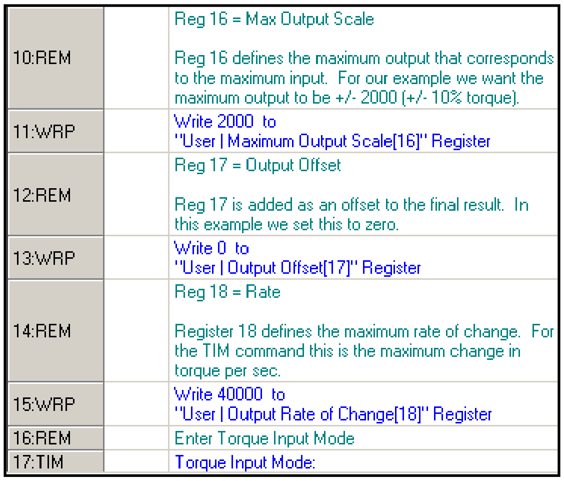
WARNING: The servo will go as fast as the commanded torque allows.

Lines 3-9: Registers 12-15 have the same setup as the above Position Input Mode example.

Line 11: Register 16 defines the maximum output that corresponds to the maximum input. For our example we want the maximum output to be +/- 2000 (+/- 10% torque).

Line 13: Set the Output Offset to zero.

Line 15: Limit the maximum change in torque per second to 40000.



10:REM		Reg 16 = Max Output Scale Reg 16 defines the maximum output that corresponds to the maximum input. For our example we want the maximum output to be +/- 2000 (+/- 10% torque).
11:WRP		Write 2000 to "User | Maximum Output Scale[16]" Register
12:REM		Reg 17 = Output Offset Reg 17 is added as an offset to the final result. In this example we set this to zero.
13:WRP		Write 0 to "User | Output Offset[17]" Register
14:REM		Reg 18 = Rate Register 18 defines the maximum rate of change. For the TIM command this is the maximum change in torque per sec.
15:WRP		Write 40000 to "User | Output Rate of Change[18]" Register
16:REM		Enter Torque Input Mode
17:TIM		Torque Input Mode: