X Y Joystick

Associated files:

- QCI-AN056_XYJoystick.qcp

Required Software

QuickControl
Rev 4.6 or higher

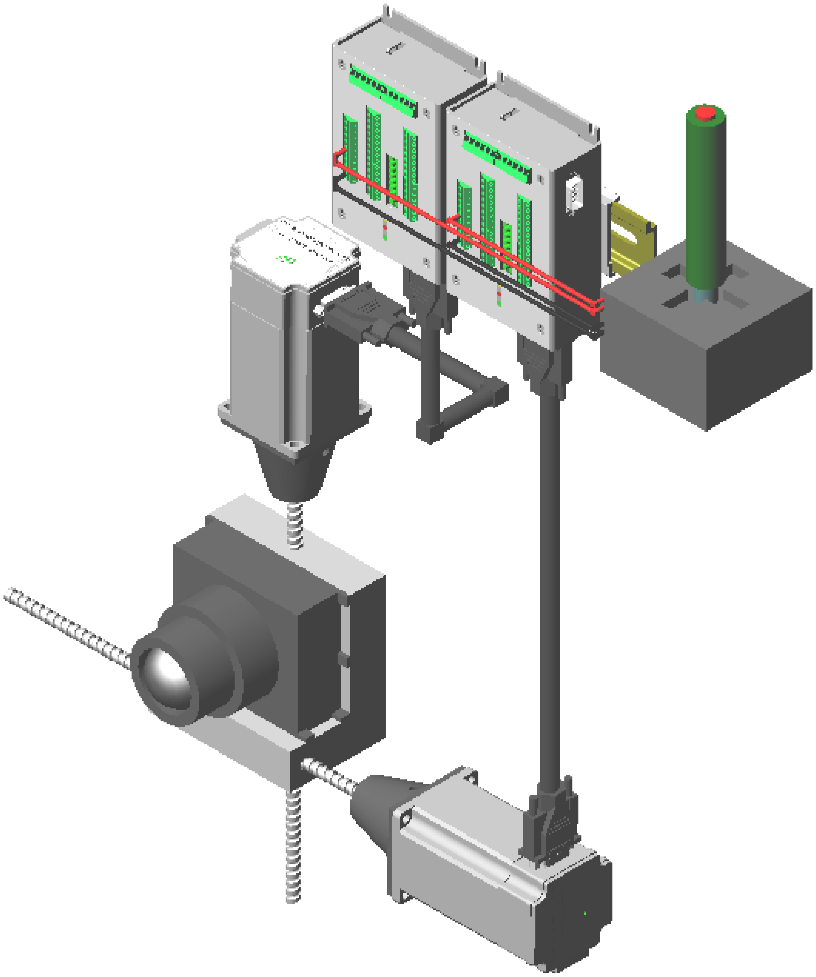
Overview

The 2-axes system uses an analog joystick to move one object on a two dimensional plane. In this example, we will be using a 2-axes joystick controlling a camera. The analog sensors in the joystick wire to the analog inputs on the QuickSilver Controllers. This allows the controller to read the analog signal.

For more information regarding Analog Inputs and Analog Reads, please refer to:
QCI-AN023_AnalogInputs
QCI-AN047_InputModeJoystick

System Parameters

This system requires:
1) 2 x Controllers, Motors and Associated Cables
2) 1 x Camera
3) 2 x Analog Sensors



QCI-AN056_XYJoystick.qcp

2-Axes Analog Homing

This homing sequence uses the Hard Stop sequence to locate to the minimum and maximum travel. This requires some initialization steps:

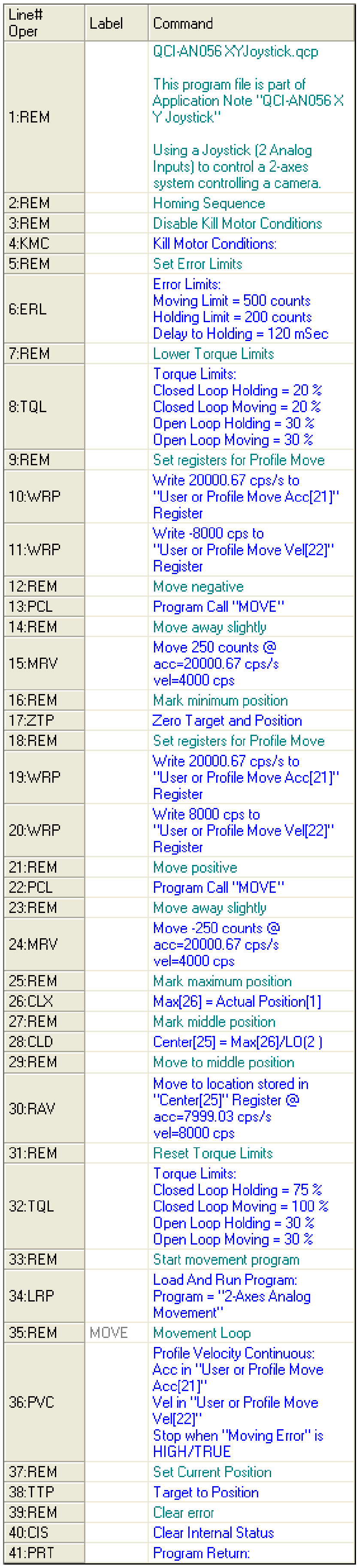
1) Disable Kill Motor Conditions – needed to keep controller from ending the program when hitting a Hard Stop.
2) Set Error Limits – needed to let the motor register sense a Hard Stop quickly.
3) Set Torque Limits – needed to prevent the motor from damaging the equipment.

The first move command is in a negative direction. This finds the lower limit then moves a small amount away before setting the actual position. To save room, like commands are in a function at the end of the program and is called when needed.

The next move is in a positive direction. This finds the upper limit. This works the same as the far positive move.

The next operation sets the normal Torque Limits then runs the 2-Axes Analog Movement and begins the movement sequence.

Refer to the QCI-AN001_HomingTechniques application note for further explanation.



Line# Oper	Label	Command
1:REM		QCI-AN056 XYJoystick.qcp This program file is part of Application Note "QCI-AN056 X Y Joystick" Using a Joystick (2 Analog Inputs) to control a 2-axes system controlling a camera.
2:REM		Homing Sequence
3:REM		Disable Kill Motor Conditions
4:KMC		Kill Motor Conditions:
5:REM		Set Error Limits
6:ERL		Error Limits: Moving Limit = 500 counts Holding Limit = 200 counts Delay to Holding = 120 mSec
7:REM		Lower Torque Limits
8:TQL		Torque Limits: Closed Loop Holding = 20 % Closed Loop Moving = 20 % Open Loop Holding = 30 % Open Loop Moving = 30 %
9:REM		Set registers for Profile Move
10:WRP		Write 20000.67 cps/s to "User or Profile Move Acc[21]" Register
11:WRP		Write -8000 cps to "User or Profile Move Vel[22]" Register
12:REM		Move negative
13:PCL		Program Call "MOVE"
14:REM		Move away slightly

15:MRV		Move 250 counts @
		acc=20000.67 cps/s
		vel=4000 cps

16:REM		Mark minimum position
17:ZTP		Zero Target and Position
18:REM		Set registers for Profile Move
19:WRP		Write 20000.67 cps/s to "User or Profile Move Acc[21]" Register
20:WRP		Write 8000 cps to "User or Profile Move Vel[22]" Register
21:REM		Move positive
22:PCL		Program Call "MOVE"
23:REM		Move away slightly

24:MRV		Move -250 counts @
		acc=20000.67 cps/s
		vel=4000 cps

25:REM		Mark maximum position
26:CLX		Max[26] = Actual Position[1]
27:REM		Mark middle position
28:CLD		Center[25] = Max[26]/LO(2)
29:REM		Move to middle position

30:RAV		Move to location stored in "Center[25]" Register @
		acc=7999.03 cps/s
		vel=8000 cps

31:REM		Reset Torque Limits
32:TQL		Torque Limits: Closed Loop Holding = 75 % Closed Loop Moving = 100 % Open Loop Holding = 30 % Open Loop Moving = 30 %
33:REM		Start movement program
34:LRP		Load And Run Program: Program = "2-Axes Analog Movement"
35:REM	MOVE	Movement Loop
36:PVC		Profile Velocity Continuous: Acc in "User or Profile Move Acc[21]" Vel in "User or Profile Move Vel[22]" Stop when "Moving Error" is HIGH/TRUE
37:REM		Set Current Position
38:TTP		Target to Position
39:REM		Clear error
40:CIS		Clear Internal Status
41:PRT		Program Return:

QCI-AN056_XYJoystick.qcp

2-Axes Analog Movement

This program uses the Position Input Mode (PIM) command to change the analog data into position data.

Refer to QCI-AN047_InputModeJoystick for an in depth description of the operation.

The first command sets an Analog Continuous Read to Analog Channel #1. All data received here records in Register 12.

Register 13 reads the current position of the joystick as the Zero Cross Over.

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

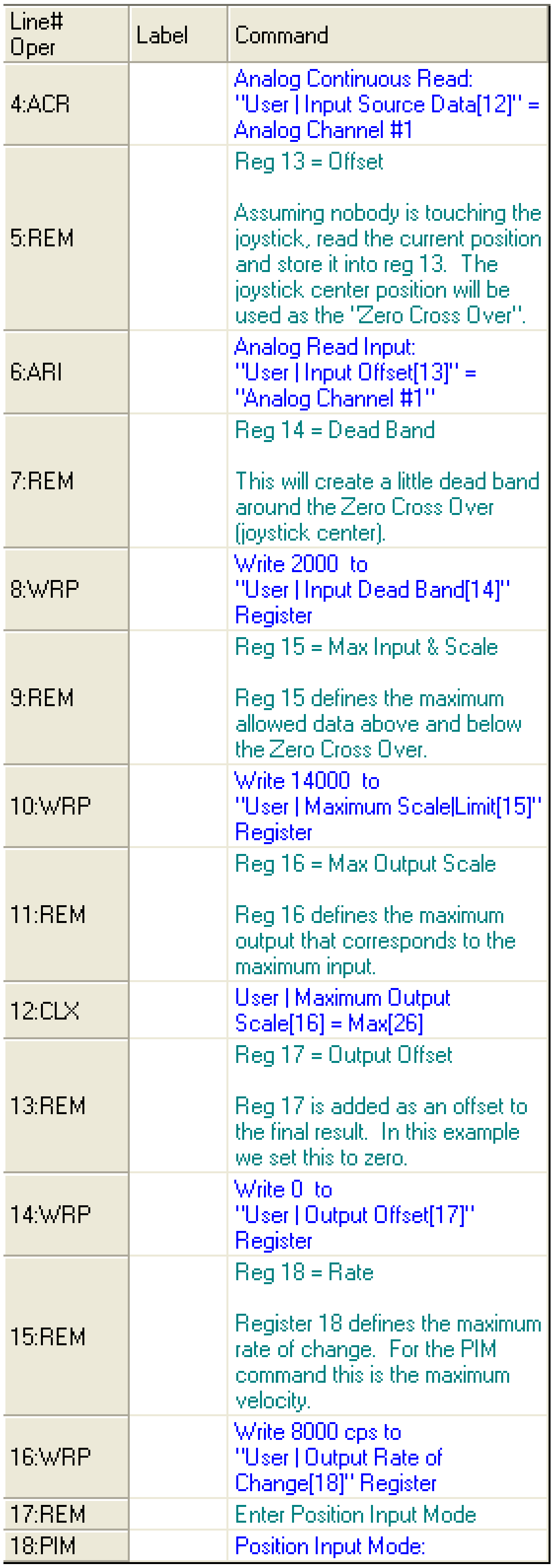
Register 15 defines the maximum allowed data above and below the Zero Cross Over.

Register 16 sets the upper and lower limits of movement.

Register 17 sets the output offset. In this example, it is zero.

Register 18 sets the maximum rate of change. In this example, it is 8000 counts per second. This limits the speed of the motor at one revolution per second.

The Position Input Mode executes and continually updates the position based on the analog input.



Line# Oper	Label	Command
4:ACR		Analog Continuous Read: "User | Input Source Data[12]" = Analog Channel #1
5: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".
6:ARI		Analog Read Input: "User | Input Offset[13]" = "Analog Channel #1"
7:REM		Reg 14 = Dead Band This will create a little dead band around the Zero Cross Over (joystick center).
8:WRP		Write 2000 to "User | Input Dead Band[14]" Register
9:REM		Reg 15 = Max Input & Scale Reg 15 defines the maximum allowed data above and below the Zero Cross Over.
10:WRP		Write 14000 to "User | Maximum Scale|Limit[15]" Register
11:REM		Reg 16 = Max Output Scale Reg 16 defines the maximum output that corresponds to the maximum input.
12:CLX		User | Maximum Output Scale[16] = Max[26]
13:REM		Reg 17 = Output Offset Reg 17 is added as an offset to the final result. In this example we set this to zero.
14:WRP		Write 0 to "User | Output Offset[17]" Register
15:REM		Reg 18 = Rate Register 18 defines the maximum rate of change. For the PIM command this is the maximum velocity.
16:WRP		Write 8000 cps to "User | Output Rate of Change[18]" Register
17:REM		Enter Position Input Mode
18:PIM		Position Input Mode: