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.
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.

<table>
<thead>
<tr>
<th>Line#</th>
<th>Label</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:ACR</td>
<td>ACR</td>
<td>Analog Continuous Read: &quot;User</td>
</tr>
</tbody>
</table>
| 5:REM | Reg13 = 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 | Reg14 = 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 | Reg15 = Max Input & Scale | Reg15 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 | Reg16 = Max Output Scale | Reg16 defines the maximum output that corresponds to the maximum input.
| 13:REM | Reg17 = Output Offset | Reg17 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 | Reg18 = 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:PIN | Position Input Mode: | |