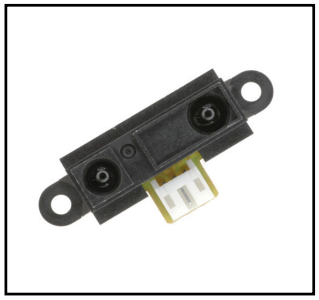
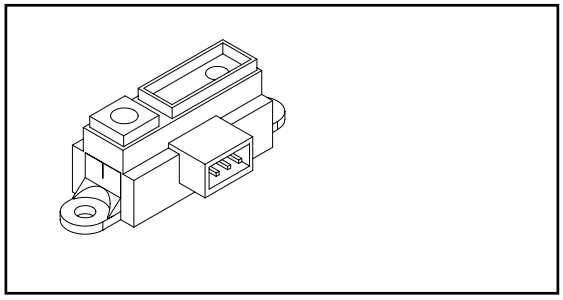
Optical Position Sensor – Sharp GP2D120XJ00F

Included Files:

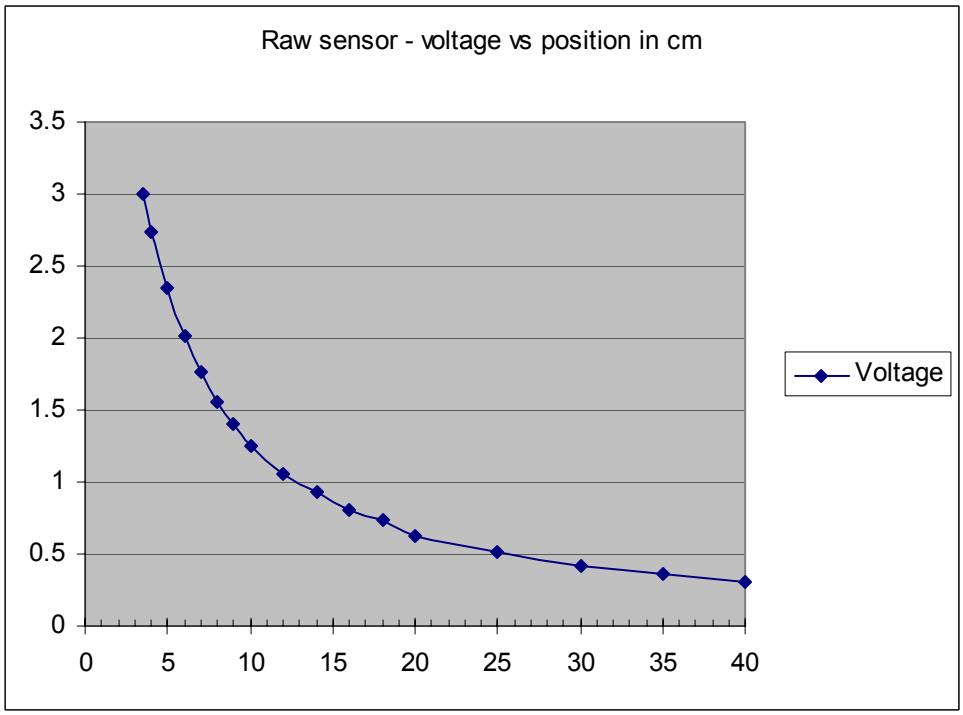
OpticalPositionSensor.qcp
OpticalPositionSensor.xls





Position sensing may be done using a dancer arm with a potentiometer or rotary hall effect sensor to determine arm position. Another method is to use optical sensing. Sharp makes the GP2D120XJ00F reflective optical sensor which has a useful range from approximately 4 to 40 centimeters. This sensor uses a near infrared LED to illuminate a diffuse reflector (designed for sheet paper). The internal processing element updates an analog output voltage approximately 30 times per second. This may be used directly to determine the material on a roll (if it has the proper optical properties) or may sense a dancer arm carrying a diffuse reflective surface. This sensor retails for less than $20, including connector.

The sensor is powered from +5v , and produces an output from approximately .3 to 3v over the range of 4 to 40 centimeters. The output is not linear, but, rather is approximately an offset reciprocal function.

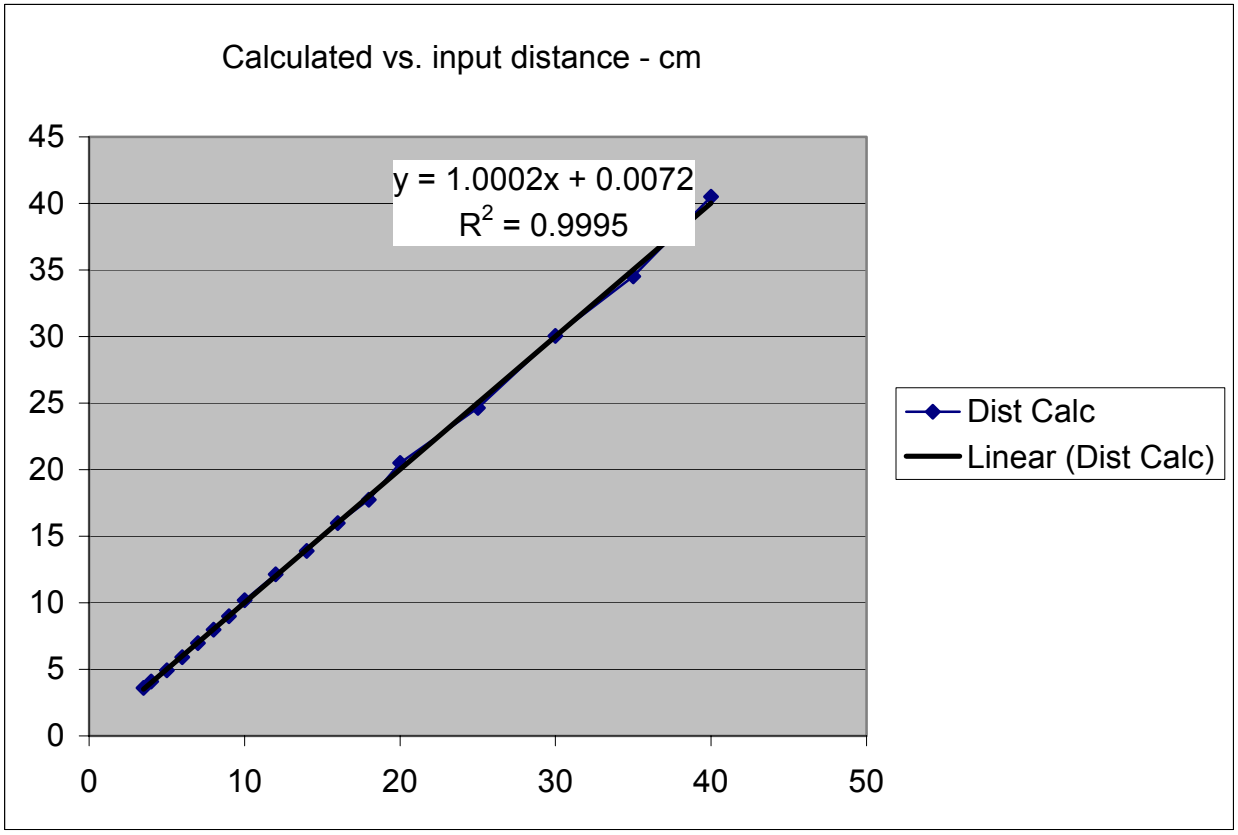


This may be linearized by the following equation, where R is distance in 0.1mm increments:

R = (m/(ADC + a)) – b
Where m=1496603, a=600, b=132

(See the included Excel file for details)

This results in the following calculated distance vs. input distance transfer:



This calculation may easily performed using a simple routine in the second thread. The example program is configured to leave its result in Register 12 for use with a VIM or PIM command.

OpticalPositionSensor.qcp

Main program

Line# Oper	Label	Command
1:KMC		Kill Motor Conditions: If Temp/Driver Enable Fault or Over Voltage TRUE
2:ERL		Error Limits: Moving Limit = 1000 counts Holding Limit = 1000 counts Delay to Holding = 120 mSec Drag Mode
3:AHC		Anti-Hunt Constants: Anti-Hunt Disabled
4:EMT		Enable Multi-Tasking
5:REM		Start second thread processing the optical position sensor
6:T2S		Start Thread 2 with Program = "Optical Sensor Processing"
7:EMT		Enable Multi-Tasking
8:REM		Set accel factor high so it gets to full ratio immediately
9:ZTG		Zero Target
10:EMD		Enable Motor Driver
11:REM		**********
12:REM	ALIGN DANCER ARM	Align the Dancer Arm Read the analog voltage from the optical position sensor. Move the motor to center the dancer arm.
13:REM		CW on the motor pays out products Tight = 900 (9cm) Loose = 1300 (13 cm) for 10 cm move, about 10 cm of product motion
14:REM		Center in operational range - about 1100 for the example
15:WRP		Write 1100 to "User or Input Offset[13]" Register
16:REM		about 1 mm dead band that is +/- 5 * .1mm
17:WRP		Write 5 to "User or Input Dead Band[14]" Register
18:REM		full scale from the center is 2 cm come up to full speed in 1.5 cm
19:WRP		Write 150 to "User or Maximum Scale|Limit[15]" Register
20:REM		pick the speed to zero out the arm in about 1 second, as it will slow down as it gets closer
21:WRP		Write 30000 cps to "User or Maximum Output Scale[16]" Register
22:WRP		Write 0 to "User or Output Offset[17]" Register
23:WRP		Write 2222222.22 cps/s to "User or Output Rate of Change[18]" Register
24:VIM		Velocity Input Mode:
25:REM		Additional code may be used here to shut down the motion

Linearization Program

Line# Oper	Label	Command
1:REM		Linearize the optical sensor
2:REM		With R in units of 0.1mm per count: R= [14966304 / (ADC + 600)]-132 This gives a curve fit (against data sheet information) of .9995x-.0002 when cm result plotted against cm input. See Winder Ratio Calculation data sheet for full details.
3:REM		Read the sensor value - Analog input 4 (on IO7)
4:ARI	LOOP	Analog Read Input: "raw ranging sensor[38]" = "Analog Channel #4"
5:REM		Add offset
6:CLD		Accumulator[10] = raw ranging sensor[38] + 600
7:REM		Make sure we don't overflow the divisor - should not happen with defined range of inputs voltages, but this fixes it just in case.
8:CLD		Accumulator[10] = Min of Accumulator[10] or 32767
9:REM		Divide into 14966304
10:CLD		Accumulator[10] = 14966304 /LO(Accumulator[10])
11:REM		Subtract off final offset
12:CLD		User or Input Source Data[12] = Accumulator[10] - 132
13:CLX		Range in .1mm[39] = User or Input Source Data[12]
14:JMP		Jump to "LOOP"
15:REM		