uM-FPU Application Note 4

Measuring Distance with the Sharp GP2D12 and GP2D120 Distance Sensors

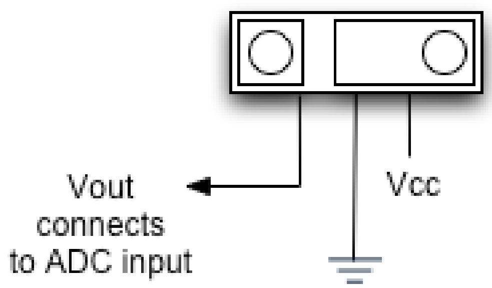
Micromega Corporation

This application note describes how to use the uM-FPU floating point coprocessor to calculate distances based on the voltage output from Sharp GP2D12 or GP2D120 distance measuring sensors.

Introduction

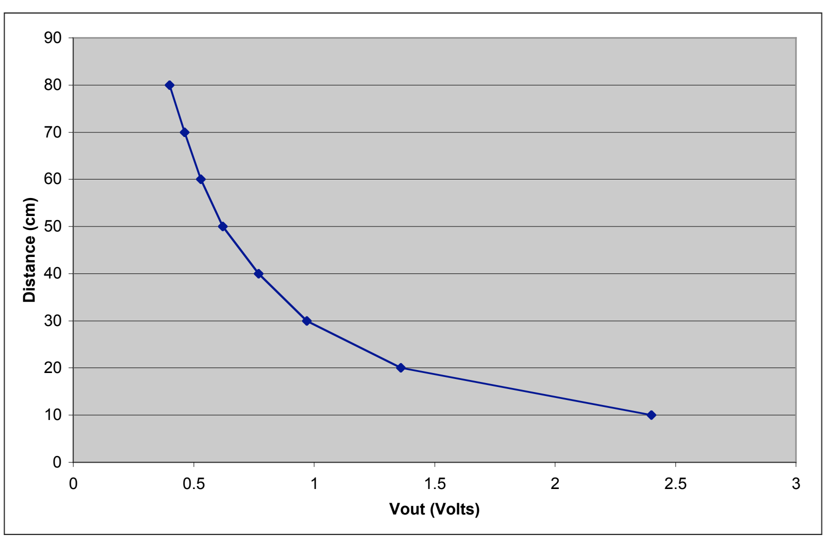
The Sharp GP2D12 and GP2D120 distance measuring sensors are easy to connect to a microprocessor through an analog-to-digital converter (ADC). Power and ground are supplied to the sensor, and an output voltage (*Vout*), proportional to the distance, is output. The *Vout* signal is connected to the ADC input.

Figure 1 - GP2D12 or GP2D120 Connection



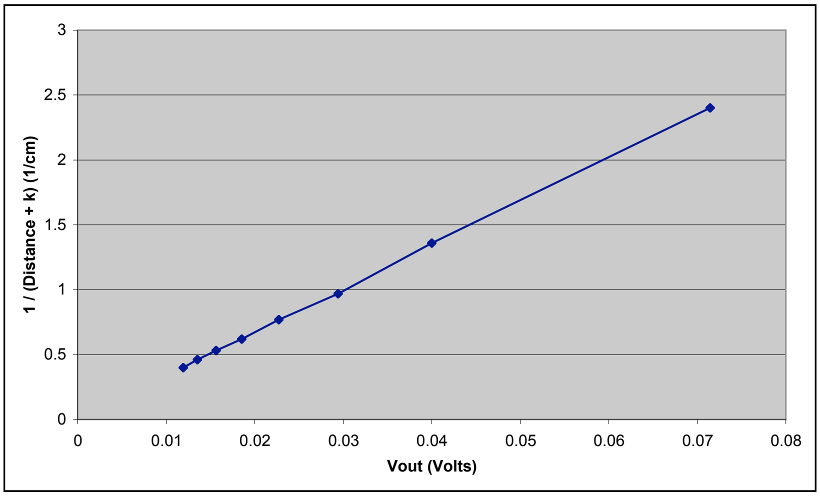
The Sharp GP2D12 is used to measure distances from 10 cm to 80 cm, and the GP2D120 is used to measure distances from 4 cm to 30 cm. Although they are easy to connect, getting the distance requires a bit more work. Looking at the graph in Figure 2 you can see that *Distance* vs *Vout* is not a straight-line relationship, so the familiar equation for a line ($y = mx + b$) is not going to work directly for calculating distance from *Vout*.

Figure 2 - Distance versus Vout (for GP2D12)



Fortunately, Sharp has defined a function that can turn the data into a straight-line relationship. Rather then look at *Distance* versus *Vout*, we look at *1/(Distance + k)* versus *Vout*. The value *k* is a constant and is equal to 4.0 for the GP2D12 and 0.42 for the GP2D120. The graph of Figure 3 shows that we now have a straight-line relationship.

Figure 3 - 1 / (Distance + k) versus Vout (for GP2D12)



By using the equation for a line ($y = mx + b$), and setting y equal to $1 / (Distance + k)$, we can rearrange the terms of the equation to get the following equation for calculating distance from *Vout*.

$$Distance = (1 / (m * Vout + b)) – k$$

The *m* and *b* values are determined by running a calibration procedure. The calibration procedure is normally run as a separate program and yields constant values for *m* and *b (*see *Calibration* Prodedure later in the document*)*. The *m*, *b* and *k* constants can be stored in uM-FPU registers as part of the initialization code in the main program. The code for calculating the distance from *Vout* is then quite straightforward.

Calculating Distance

The uM-FPU code for calculating distance from *Vout* is shown below. It assumes the values for *m*, *b* and *k* have been stored in uM-FPU registers as part of the initialization code. In this example, the adcVal is a 16-bit value loaded with the `LOADWORD` instruction (used for 10-bit, 12-bit and 16-bit ADCs). If an 8-bit ADC was used, the `LOADBYTE` instruction could be used instead.

Microprocessor Variables

```
adcVal                      ; 16-bit ADC value for Vout
```

uM-FPU Register Definitions

```
M                           ; m constant
B                           ; intercept of line (b)
K                           ; k constant
Distance                    ; computed distance
```

Distance = (1 / (M * adcVal + B)) – K

```
SELECTA+Distance            ;
LOADWORD                    ; load adcVal to register 0 and
adcVal (high byte)          ;  convert to floating point
adcVal (low byte)
```

```
FSET                    ; Distance = adcVal
FMUL+M                  ; Distance = Distance * M
FADD+B                  ; Distance = Distance + B
INVERSE                 ; Distance = 1 / Distance
FSUB+K                  ; Distance = Distance – K
```

Result

The uM-FPU *Distance* register now contains the distance in cm.

Calibration Procedure

The calibration procedure consists of measuring *Vout* at various known distances, then determining the trend line for the calibration data. The *Vout* signal is applied to an ADC to obtain a digital value. The digital value depends on the characteristics of the ADC, but by using the same ADC setup for the calibration procedure and the main program, the value of *m* and *b* for the trend line will be calculated appropriately.

For the GP2D12, *Vout* samples are taken every 10 cm from 10 to 80 cm. For the GP2D120, *Vout* samples are taken every 5 cm from 5 to 30 cm. For each sample, the data points are stored as follows: *x =Vout, y = 1 /(Distance + k)*. A spreadsheet program such as Microsoft Excel can be used to perform a trend line analysis on the data points to determine the *m* and *b* values, but the uM-FPU can also perform the trend line analysis. A sample program is provided that implements a calibration routine for the GP2D12 and GP2D120 distance sensors.

Analog-to-Digital Converter and Sensor Accuracy

Distance sensors are typically not read at a rate of more than a few samples per second, so the performance characteristics of most ADCs will be sufficient. Assuming that the noise on the Vout input signal has been kept to a minimum, the main concern is to ensure that the number of bits used for the ADC output is sufficient for the desired resolution.

If you refer to Figure 2 you can see that the change in voltage from 70 cm to 80 cm is only about 0.06 V, which corresponds to 0.006 V/cm. If you use an 8-bit ADC with a reference voltage of 5V, each bit of the ADC output represents 0.0195 V which means a one bit swing in the ADC output will result in a distance swing of about 3 cm.

The maximum voltage output from a GD2D12 sensor is about 3V. If the reference voltage for the 8-bit ADC is changed to 3V, each bit of the ADC output represents 0.0117 V, which means a one bit swing in the ADC output will still result in a distance swing of about 2 cm.

The resolution is better at shorter distances because there is a larger voltage change. Referring to Figure 2 you can see that the change in voltage from 10 cm to 20 cm is about 1V, which corresponds to 0.1 V/cm.

The following chart provides some examples of the limitations on accuracy for various combinations of ADCs and reference voltage.

ADC bits	**Reference Voltage (V)**	**V/bit**	**cm/bit (10 to 20 cm)**	**cm/bit (70 to 80 cm)**
8	5	0.0195	0.195	3.25
8	3	0.0117	0.117	1.95
10	5	0.0049	0.049	0.81
12	5	0.0012	0.012	0.20
16	5	0.000076	0.00076	0.01

If an 8-bit ADC is used, the resolution at longer distances will be less than 1 cm/bit. You will need to use at least a 10-bit ADC to get resolution of better than 1 cm/bit across the full distance range.

Further Information

Sample program that implement a calibration routine for the GP2D12 and GP2D120 distance sensors are available for various microcontrollers.

Check the Micromega website at www.micromegacorp.com for up-to-date information.

For more information on the trend line calculation used in the calibration procedure see:

uM-FPUApplication Note 3 – Calculating Trend Lines