Using the ADXL202 Duty Cycle Output

by Harvey Weinberg

Calculation of the Acceleration Output

Acceleration experienced by the ADXL202 may be calculated using the following formula:

$$Acceleration\,(in\,\text{g}) = \frac{Duty\,Cycle - Duty\,Cycle\,at\,0\,\text{g}}{Duty\,Cycle\,per\,\text{g}}$$

As outlined in the data sheet, the nominal duty cycle output of the ADXL202 is 50% at 0 *g* and 12.5% duty cycle change per *g*. Therefore to calculate acceleration from the duty cycle:

$$Acceleration\,(in\,\text{g}) = \frac{(T1/T2) - 50\%}{12.5\%}$$

If the 0 *g* duty cycle output of the ADXL202 is other than 50%, and/or the duty cycle changes more or less than 12.5% per *g*, the acceleration calculation will be inaccurate. In practice, the 0 *g* output and the sensitivity of the ADXL202 vary somewhat from device to device (see the data sheet for details). So this formula can only be used for low accuracy measurements. For higher accuracy measurements, the actual offset and scale values must be substituted.

In addition, the result of this equation would be a number in the range of ±2 for an ADXL202. In general, it is inconvenient to use real numbers in calculations with a small microcontroller since floating point mathematics would be required to get meaningful results. A preferred method for calculation of acceleration with a small microcontroller using fixed point math is included in this application note, along with two simple methods of calibration to find the actual offset and scale values.

This application note outlines methods to decode the duty cycle output, conversion from duty cycle to acceleration (or tilt angle), and calibration of the ADXL202. These methods are geared toward use with 8-bit microcontrollers having limited computational capability.

Decoding the Output

The most direct way to decode the duty cycle output is shown in Figure 1. A counter is started at the rising edge of the X output (Ta = 0). The count at the falling edge (Tb) is recorded, and the timer is stopped at the next rising edge of the X output (Tc). This process is then repeated for the Y output (Td, Te, and Tf).
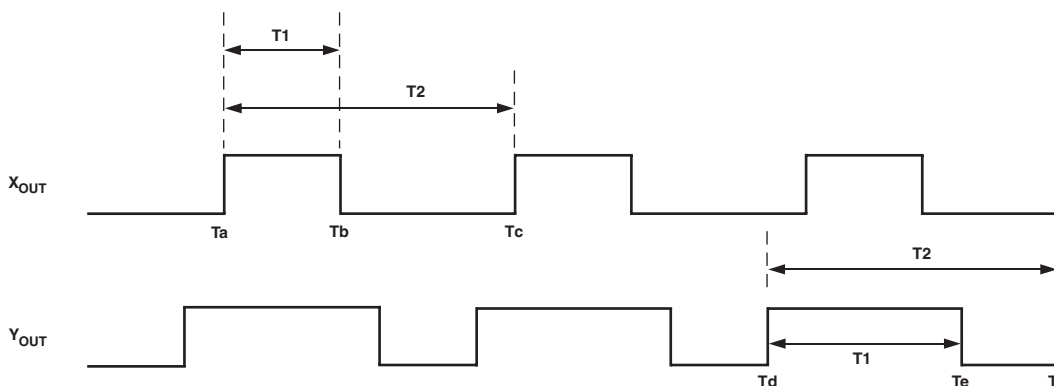

Figure 1. Basic Decode Technique for the ADXL202

While this technique is very easy to understand, you can only acquire one sample of acceleration from both axes every three cycles (i.e., $3 \times T2$) since you must wait for the next rising edge of X_{OUT} after Tf.

Note that while T2 is nominally constant, it does change over temperature and contains some jitter. For systems that do not require resolutions of better than 100 mg, T2 may be measured only once. For more accurate measurements, several T2 measurements should be made and averaged. The average should be updated periodically to account for T2 drift over temperature.

An Improved PWM Decode Scheme

Since the duty cycle modulator (DCM) uses the same triangle wave reference for the X and Y channels, the midpoints of the T1 of each period must be coincident. This is illustrated in Figure 2. Here it is shown that even though the X and Y duty cycle outputs are different, the midpoints of T1 are synchronized. Therefore, an improved PWM decode technique can be used to speed up the data acquisition time. Figure 2 shows the sequence of events. A counter is started at the rising edge of the X output (Ta = 0). The count at the falling edge of the X output (Tb) is recorded. Then the counts at the rising and falling edges of the Y output (Tc and Td) are recorded. A flow chart outlining this method of calibration is shown in Figure 3.

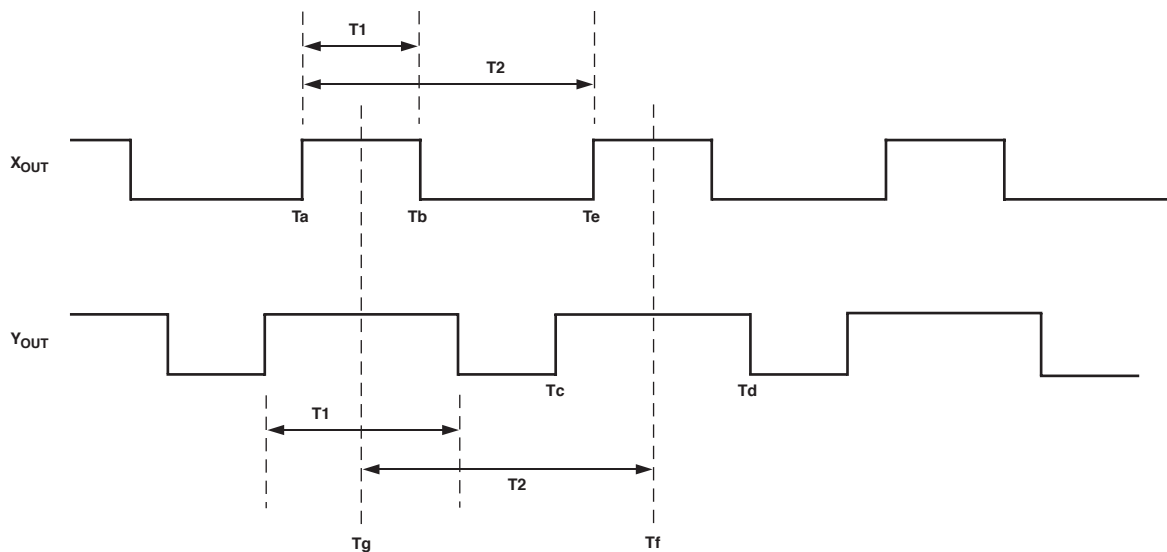*Figure 2. High Speed Decoding Technique for the ADXL202*

By definition: $T1x = Tb - Ta = Tb$ (if the counter is zero at Ta)

$$T1y = Td - Tc$$

$$T2x = T2y = Te - Ta = Tg - Tf$$

Since the midpoints of the high states of the X and Y duty cycle signals are coincident:

$$T2 = [Td - ((Td - Tc)/2)] - [(Tb - Ta)/2]$$

$$T2 = [Td - ((Td - Tc)/2)] - [Tb/2] \quad \text{(if the counter is zero at } Ta)$$

The advantages of this system of decoding are:
1. One sample of acceleration from both axes can be acquired every two $T2$ cycles.
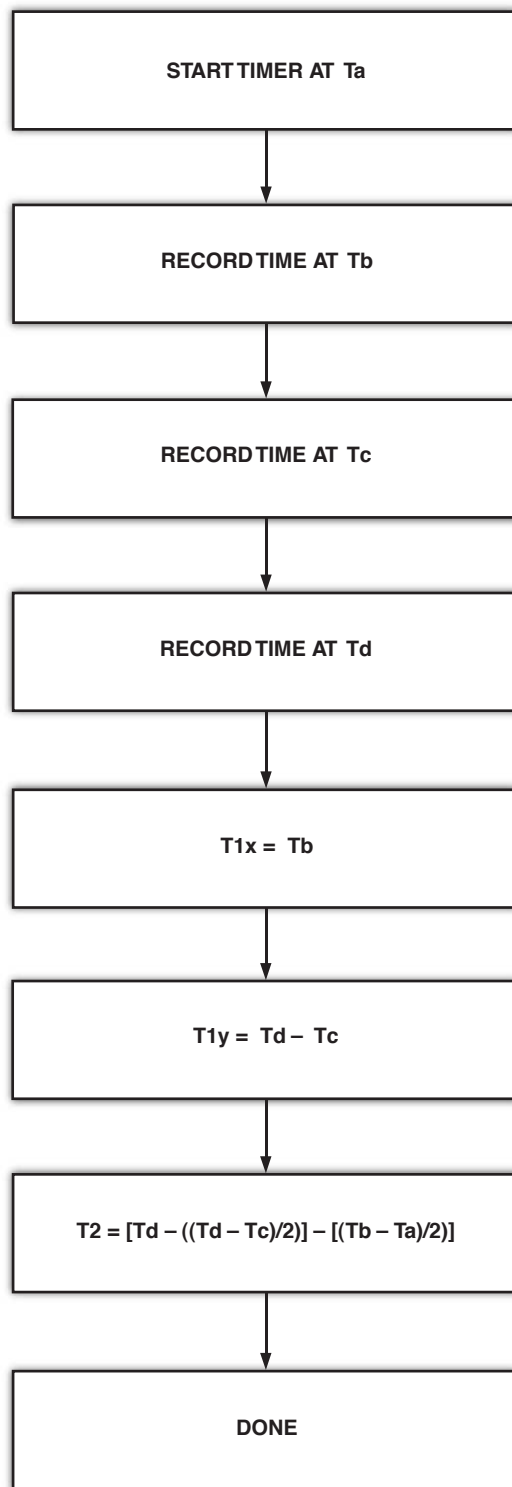2. $T2$ is only calculated once for both the X and Y signals.

START TIMER AT Ta

RECORD TIME AT Tb

RECORD TIME AT Tc

RECORD TIME AT Td

T1x = Tb

T1y = Td – Tc

T2 = [Td – ((Td – Tc)/2)] – [(Tb – Ta)/2)]

DONE

Figure 3. Flow Chart for High Speed Decoding Technique for the ADXL202

AN-604

Calibration of the ADXL202

The easiest way to calibrate the ADXL202 is by using the earth's gravity as a reference input. If your application does not require high accuracy, you may use a "quick calibration" technique. Here we assume that the sensitivity of the accelerometer is 12.5% duty cycle per *g* (the typical specification). One digital input may be used to tell the microcontroller that it is in calibrate mode. When in calibrate mode, the accelerometer must be level, with the X and Y axes horizontal to the earth so both axes experience 0 *g*.

When the microcontroller is told to calibrate, the microcontroller reads the duty cycle output (T1) and period (T2) of the accelerometer from each axis. Several readings of T1 and T2 may be averaged to improve accuracy. These values are stored as calibration constants and retained for use in calculating the acceleration after calibration. A scale factor (K) used to scale the acceleration output into an n-bit word should also be calculated. The calibration constants for each axis are:

1. T2cal = the value (from the microcontroller's counter) of T2 during calibration. T2cal must be stored, as the value of T2 does drift over temperature and has jitter.

2. Zcal = the value (from the microcontroller's counter) of T1 during calibration.

3. Bit scale factor. You choose the bit scale factor to determine the resolution (in bits) of the acceleration calculation.

4. K = [4 × (T2cal × bit scale factor)/T2cal]
   Note that K need only be calculated once for the two axes if you choose the same bit scale factor for both axes.

The bit scale factor will be determined by the size (in bits) of the result desired. For example, if a result of ±1 *g* = ±128 counts (to result in an 8-bit number) is needed, the scale factor would be 256. For a result of ±1 *g* = ±90 counts (to approximate 1° per count), the scale factor would be 180.

Calculation of Acceleration from Duty Cycle

Once the calibration constants are known, only two formulas are required for the calculation of acceleration. They are:

$$Zactual = \frac{Zcal \times T2actual}{T2cal}$$

Where *T2actual* is the current measurement of *T*2. This formula corrects the 0 *g* value for changes in *T*2 due to drift or jitter.

$$Acceleration = \frac{K \times (T1 - Zactual)}{T2actual}$$

Note that all of the operations are fixed point, and therefore fairly easy and fast for any microcontroller to perform.

Care should be taken in the correct ordering of operations to preserve resolution. When designing software, keep an eye out for ways to order mathematical operations that will not drop significant bits. In general, perform addition, multiplication, subtraction, and division in that order. Double and triple precision fixed point arithmetic (16- and 24-bit) will be required in most applications. Algorithms for multibyte fixed point arithmetic are widely available in application notes published by microcontroller manufacturers.

High Accuracy Calibration Method for the ADXL202

Variance in sensitivity (i.e., duty cycle % per g) from part to part is normal and will result in a small error if not accounted for. If your application requires high accuracy measurements, the rotational calibration method presented below is preferred.

Simply place the ADXL202 such that the Z axis is horizontal and rotate the ADXL202 over at least 360°. The ADXL202 will then be exposed to ±1 g in both the X and Y axes. The rotation must be done slowly in order to minimize the effects of centrifugal acceleration.

The following pieces of calibration data must be retained for use in calculating the acceleration after calibration:

1. T2cal. That is, the value of T2 during the calibration procedure.

2. *Zcal*. That is, the 0 g value of $T1$ at the time of calibration. It is calculated as follows:

$$Zcal = \frac{T1max - T1min}{2}$$

3. Bit scale factor. You choose the bit scale factor to determine the resolution (in bits) of the acceleration calculation.

4. *K*. Where *K* is the scale factor. *K* may be calculated as follows:

$$K = \frac{T2cal \times bit\ scale\ factor}{T1max - T1min}$$

The bit scale factor used here is the same as outlined in the quick calibration technique.

Variance in part-to-part sensitivity is corrected in the denominator by calculating the actual duty cycle output obtained from exposure to 2 g of acceleration (±1 g from gravity).

A flow chart outlining a calibration algorithm is shown in Figure 4. One digital input is used to tell the microcontroller that it is in calibrate mode.

Calculation of the acceleration is done as presented above. That is:

$$Acceleration = \frac{K \times (T1 - Zactual)}{T2actual}$$

Where:

$$Zactual = \frac{Zcal \times T2actual}{T2cal}$$

For maximum accuracy, several readings of $T2$ should be averaged to reduce error induced by $T2$ jitter. The flow chart (Figure 4) does not include this operation. It is up to the user to determine how many averaged samples of $T2$ are suitable in the application. Generally four to eight averaged samples of $T2$ should be sufficient for systems with resolutions of 1 mg or greater.
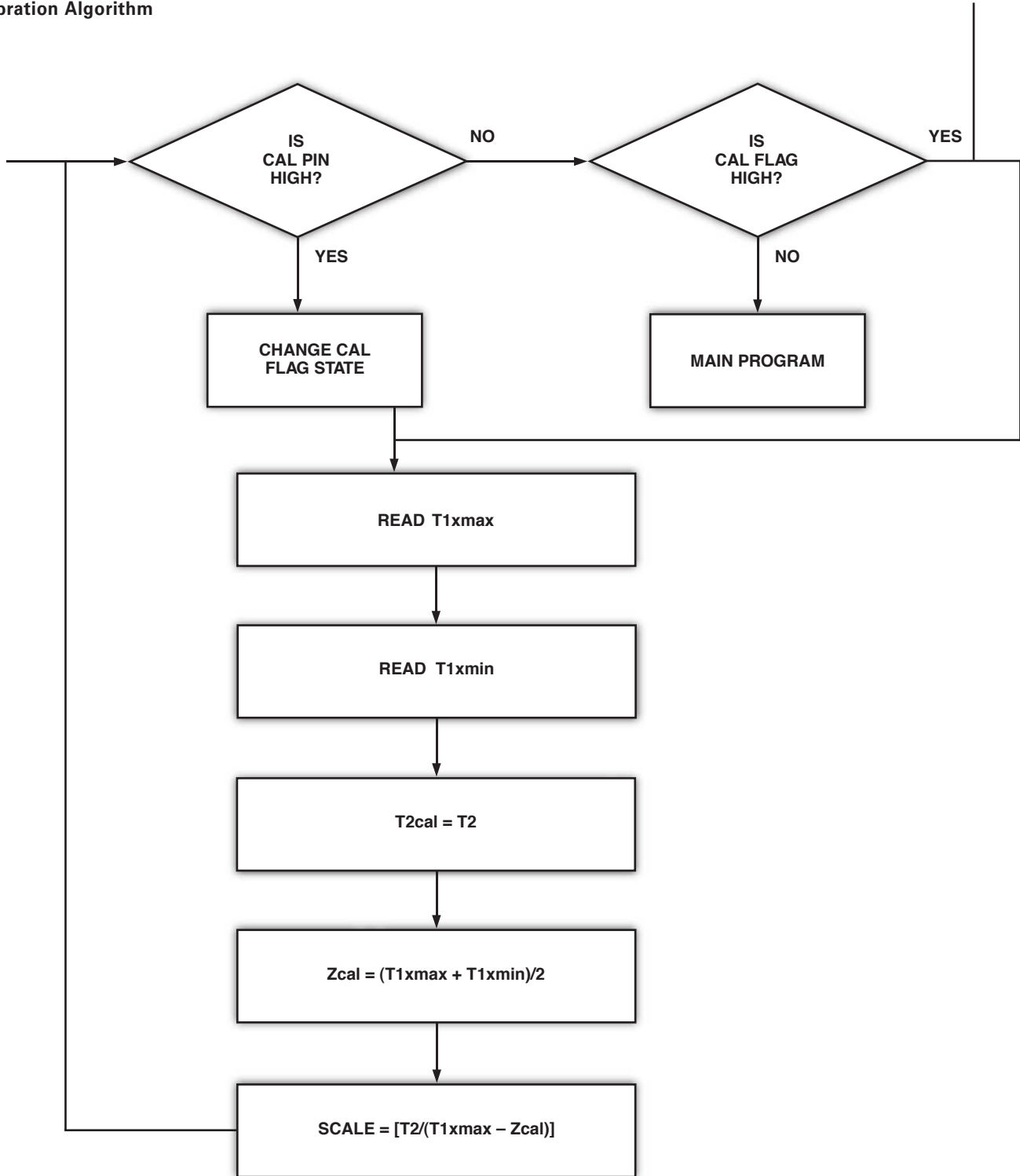
Calibration Algorithm

Figure 4. Flow Chart for the ADXL202 Calibration Algorithm