TECHNICAL NOTE

ONE TECHNOLOGY WAY • P.O. BOX 9106 • NORWOOD, MASSACHUSETTS 02062-9106 • 781/329-4700

USING THE ADXL202/ADXL210 WITH THE PARALLAX™ BASIC STAMP MODULE TO SPEED ALGORITHM DEVELOPMENT

by Harvey Weinberg

Introduction

The ADXL202/210 are low cost, complete 2 axis accelerometers with pulse width modulated outputs that can measure both static acceleration (e.g. tilt or gravity) and dynamic acceleration (e.g. vibration). Most applications for the ADXL202/210 require a small, 8 bit microcontroller. While microcontrollers are terrific low cost signal processing tools, they usually need to be programmed using Assembly or some other low-level language. Algorithm development and prototyping however, are most conveniently done using a high level language.

The Parallax Basic Stamp module is small, low cost, general purpose I/O computer that is programmed in a simple form of BASIC. Two types of Basic Stamps are available differing mainly in the number of I/O and amount of memory available. The pulse width modulated output of the ADXL202/10 can be read directly by either of the Basic Stamp modules, so no A/D converter is necessary.

In this technical note, we will discuss how to use the ADXL202/210 with the Basic Stamp as an efficient method of algorithm development.

Module Selection

While both the Basic Stamp 1 and Basic Stamp 2 modules will work with the ADXL202/210, the Basic Stamp 2 is the module of choice. It has an enhanced Basic instruction set that facilitates it's use with the ADXL202/210. The additional I/O and memory space are also advantageous.

Basic Stamp 2 Interface: Hardware Connection

The easiest way to go is to use the Basic Stamp carrier board with an ADXL202EB (or ADXL210EB) as shown in Figure 1. Here a Basic Stamp 2 and an ADXL202EB is shown along with a piezo buzzer used in the application. This configuration (carrier board and ADXL202EB) allows easy connection of the Basic Stamp to the PC for programming and debugging, and relieves the user from having to solder any SMT parts. The schematic is shown in Figure 2.

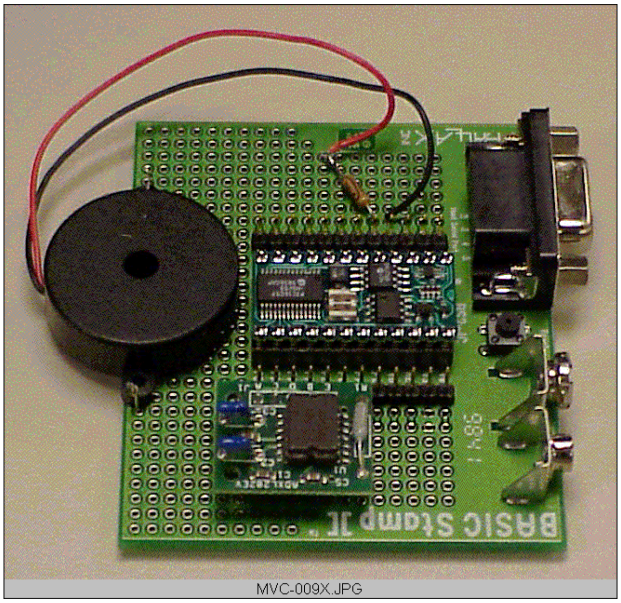

Figure 1. Basic Stamp 2 Carrier Board and ADXL202EB

In order to minimize external components and connections, power and ground for the ADXL202/210 is provided via Basic Stamp I/O. While this is convenient, it is not the lowest noise method of operation. For applications where very low noise is required, separate power and ground connections to the ADXL202/210 are recommended

9/15/99, Rev. 0

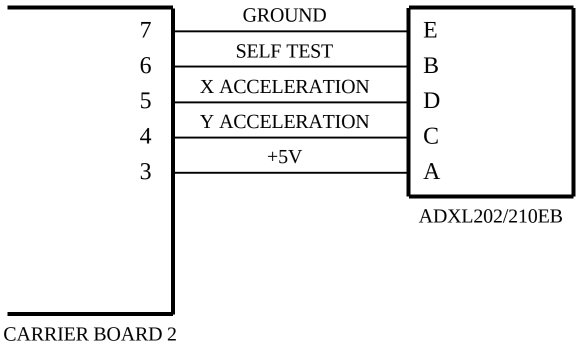

Figure 2. Schematic for Basic Stamp 2 and ADXL202EB Interface

Software Interface
Normally, one would be interested in measuring the T1 and T2 output from the accelerometer in order to determine the duty cycle (and the acceleration) output of the accelerometer. Use the software routine below to measure T1x, T1y, and T2:

```
FREQ    VAR     WORD  ‘variable declarations
T1x     VAR     WORD  ‘T1 X channel
T1y     VAR     WORD  ‘T1 Y channel
T2      VAR     WORD  ‘PWM period

LOW     7             ‘PIN 7=GROUND
HIGH    3             ‘PIN 3=+5V
INPUT   4             ‘Ax IN
INPUT   5             ‘Ay IN
LOW     6             ‘SELF TEST OFF

COUNT 8,500,FREQ      ‘count how many cycles
                      ‘in 500ms
T2=25000/(FREQ/20)    ‘T2 is the period in µs

T1_LOOP
PULSIN 5,1,T1y  ‘read T1y
T1y=2*T1y             ‘convert to µs
PULSIN 4,1,T1x  ‘read T1x
T1x=2*T1x             ‘convert to µs
```

The first section simply declares variables that will be used later. The variables are set up as 16 bit to avoid overflow. The second section sets up the I/O to supply power and ground to the ADXL202/210. The third section performs the actual measurements of T1 and T2. T2 is measured by counting how many pulses occur in 500ms. While this appears slow, it doesn’t affect performance in the real world as it is only necessary to read T2 from time to time (say every minute or so) since T2 is fairly stable. Note that T2 need only measured on one channel.

T1 measurements may be looped at whatever speed is appropriate for the application. The T1 loop starts at the T1_LOOP label.

Once T1x, T1y, and T2 are known, acceleration can be calculated very simply.
The nominal scale factor of the ADXL202 is 12.5% / *g* and 4% / *g* for the ADXL210 (see the ADXL202/210 data sheet for details). To read the acceleration directly from T1x, T1y, and T2:
For the ADXL202
X axis acceleration = 8 * T1x / T2
Y axis acceleration = 8 * T1y / T2

For the ADXL210
X axis acceleration = 25 * T1x / T2
Y axis acceleration = 25 * T1y / T2

Basic Stamp 1 Interface:
Hardware Connection
The hardware connection of a Basic Stamp 1 is very similar to that of the Basic Stamp 2, and is shown in Figure 3. Because of differences in the carrier boards, (Basic Stamp 1 versus Basic Stamp 2) the ADXL202/210 hardware interface uses different I/O pins. This difference in reflected in the software interface.

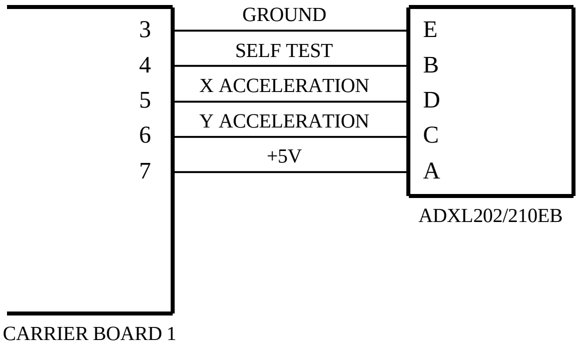

Figure 3. Schematic for Basic Stamp 1 and ADXL202EB Interface

Software Interface
As noted earlier, the instruction set of the Basic Stamp 1 is a subset of the Basic Stamp 2 instruction set. Most significantly for this application, the Basic Stamp 1 does not have a COUNT instruction. So T2 is calculated by adding the positive and negative pulse width of a few cycles from one channel.

```
T1x     VAR     WORD  ‘variable declarations
T1y     VAR     WORD
T2      VAR     WORD
TEMP    VAR     WORD
CNT     VAR     BYTE
```

```
LOW     3                   ‘PIN 7=GROUND
HIGH    7                   ‘PIN 3=+5V
INPUT   5                   ‘Ax IN
INPUT   6                   ‘Ay IN
LOW     4                   ‘SELF TEST OFF

T2=0                        ‘find T2
FOR CNT=1 TO 10
PULSIN 5,1,T1x
PULSIN 5,0,TEMP
T2=T2+TEMP
T2=T2+T1x                   ‘read T2
NEXT


T1_LOOP
PULSIN 5,1,T1x   ‘read T1x
PULSIN 6,1,T1y   ‘read T1y
T1x=T1x*10
T1y=T1y*10
```

Again the first section declares variables that are used later in the program, the second section sets up the I/O to run the ADXL202/210EB, the third section calculated, and the last section (starting from T1_LOOP) reads the T1 value for the X and Y channel.

The Basic Stamp 1 counts in 10µs increments. Since the T2 routine adds up 10 T2 periods, the result is directly given in µs. T1x and T1y are multiplied by 10 to have their results in µs as well.

The actual acceleration may be computed using the equations described in the Basic Stamp 2 Software Interface section.

Conclusion

Using a high level language such as BASIC for algorithm development is generally much easier than coding in a low level language. The ADXL202/210EB mated to the Basic Stamp carrier board makes a fast and powerful development tool as it lets the user concentrate on developing their algorithm rather than dealing with Assembly code.